3GPP TSG CN Plenary Meeting #25 **NP-040396**
8th – 10th August 2004 Palm Springs, US.

Source: TSG CN WG4

Title: Corrections on IMS Rel-5

Agenda item: 8.1

Document for: APPROVAL

Spec	CR	Rev	Doc-2nd-Level N4-04	Phase	Subject	Cat	Ver_C
29.228	123	2	1161	Rel-5	Simplification of the User Profile Split concept	F	5.8.0
29.228	124	2	1162	Rel-6	Simplification of the User Profile Split concept	A	6.3.0
29.229	057		0965	Rel-5	Simplification of the User Profile Split concept	F	5.7.0
29.229	058		0966	Rel-6	Simplification of the User Profile Split concept	A	6.1.0

3GPP TSG CN WG4 Meeting #24 **N4-040965**
Sophia Antipolis, FRANCE, 16th – 20th AUGUST 2004

CR-Form-v7

CHANGE REQUEST

⌘ **29.229** **CR** **057** ⌘ **rev** **-** ⌘ Current version: **5.7.0** ⌘

*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 **X**

Title:	⌘	Simplification of the User Profile Split concept		
Source:	⌘	CN4		
Work item code:	⌘	IMS-CCR	***Date:*** ⌘	23/06/2004
Category:	⌘	**F**	***Release:*** ⌘	Rel-5

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:
2 *(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)*

Reason for change: ⌘

This is an Essential Correction

Extensive discussion has taken place on the relative benefits and problems with the concept of splitting the user profile into Registered, Unregistered and Common parts. The user profile split was brought into the definition of the Cx interface at an early stage as a mechanism to allow for the otpimisation of the Cx interface in terms of
1. reducing the amount of data that is passed across it,
2. reducing the memory requirements placed on the S-CSCF, and
3. reducing the processing load in the S-CSCF.

However, because of the option to download parts of or the entire of the user profile, and because the S-CSCF should not have any concept of which part of the profile it actually has, with further consideration it has become clear that a number of significant problems exist.

- when performing a STORE_SERVER_NAME de-registration type which results in the subscriber being in the unregistered state but the S-CSCF has only the registered part of the User Profile, and has no mechanism to retrieve the unregistered part of the user profile.
- When a profile push occurs, there is only a requirement to send the part (ie the registered part or the unregistered part) of the user profile to the S-CSCF. However, if the S-CSCF currently holds the entire profile, the S-CSCF is obliged to overwrite that information with the part it receives and hence loses some subscriber information.
- It is currently not specified how the S-CSCF knows which parts of the user profile belong to the registered part or unregistered part of the service profile.
- It is not defined which values of the User-Data-Request-Type AVP are allowed in the cases when it's value is relevant; it is not e.g. sensible for the

	S-CSCF to ask for the unregistered profile part if the user is registered. Because of the increasingly complex solutions that are needed to fully solve all of these problems and maintain the concept of the user profile split, CN4 has become stalled in its discussions. For that reason, a completely alternative solution is considered: - to remove the option to download parts of the user profile altogether, and so only allow the complete profile to be downloaded, and - to add an indication to the IFC indicating the part to which it belongs. With this alternative solution the problems identified above are no longer in existance.
Summary of change: ⌘	The concept of downloading parts of the user profile is removed from the specs. An explicit identification of whether an IFC is related to the registered or unregistered subscriber state is added, allowing S-CSCF's to optimise their processing cycles upon receipt of SIP messages
Consequences if not approved: ⌘	The problems identified in 'Reasons for Change' continue. This leads to continuing problems from misalignment between the User Profile part stored at the S-CSCF and the user state, resulting in SIP messages being incorrectly processed and hence, subscribers experiencing incorrect behaviour from the network.

Clauses affected:	⌘	6.1.3, 6.3, 6.3.25			
Other specs affected:	⌘	**Y** / **N** **X** /	Other core specifications	⌘	29.228 CR 123
		/ **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.

6.1.3 Server-Assignment-Request (SAR) Command

The Server-Assignment-Request (SAR) command, indicated by the Command-Code field set to 301 and the 'R' bit set in the Command Flags field, is sent by a Diameter Multimedia client to a Diameter Multimedia server in order to request it to store the name of the server that is currently serving the user.

Message Format

```
<Server-Assignment-Request> ::= < Diameter Header: 301, 167772151, REQ, PXY >
                                < Session-Id >
                                { Vendor-Specific-Application-Id }
                                { Auth-Session-State }
                                { Origin-Host }
                                { Origin-Realm }
                                [ Destination-Host ]
                                { Destination-Realm }
                                [ User-Name ]
                                *[ Public-Identity ]
                                { Server-Name }
                                { Server-Assignment-Type }
~~{ User-Data-Request-Type }~~
                                { User-Data-Already-Available }
                                *[ AVP ]
                                *[ Proxy-Info ]
                                *[ Route-Record ]
```

****** *Next Changed Section* ******

6.3 AVPs

The following table describes the Diameter AVPs defined for the Cx interface protocol, their AVP Code values, types, possible flag values and whether or not the AVP may be encrypted. The Vendor-Id header of all AVPs defined in this specification shall be set to 3GPP (10415).

Table 6.3.1: Diameter Multimedia Application AVPs

				AVP Flag rules				
Attribute Name	**AVP Code**	**Section defined**	**Value Type**	**Must**	**May**	**Should not**	**Must not**	**May Encr.**
Visited-Network-Identifier	1	6.3.1	OctetString	M, V				No
Public-Identity	2	6.3.2	UTF8String	M, V				N
Server-Name	3	6.3.3	UTF8String	M,V				No
Server-Capabilities	4	6.3.4	Grouped	M, V				No
Mandatory-Capability	5	6.3.5	Unsigned32	M, V				No
Optional-Capability	6	6.3.6	Unsigned32	M, V				No
User-Data	7	6.3.7	OctetString	M, V				No
SIP-Number-Auth-Items	8	6.3.8	Unsigned32	M, V				No
SIP-Authentication-Scheme	9	6.3.9	UTF8String	M, V				No

SIP-Authenticate	10	6.3.10	OctetString	M, V				No
SIP-Authorization	11	6.3.11	OctetString	M, V				No
SIP-Authentication-Context	12	6.3.12	OctetString	M, V				No
SIP-Auth-Data-Item	13	6.3.13	Grouped	M, V				No
SIP-Item-Number	14	6.3.14	Unsigned32	M, V				No
Server-Assignment-Type	15	6.3.15	Enumerated	M, V				No
Deregistration-Reason	16	6.3.16	Grouped	M, V				No
Reason-Code	17	6.3.17	Enumerated	M, V				No
Reason-Info	18	6.3.18	UTF8String	M, V				No
Charging-Information	19	6.3.19	Grouped	M, V				No
Primary-Event-Charging-Function-Name	20	6.3.20	DiameterURI	M, V				No
Secondary-Event-Charging-Function-Name	21	6.3.21	DiameterURI	M, V				No
Primary-Charging-Collection-Function-Name	22	6.3.22	DiameterURI	M, V				No
Secondary-Charging-Collection-Function-Name	23	6.3.23	DiameterURI	M, V				No
User-Authorization-Type	24	6.3.24	Enumerated	M, V				No
~~User-Data-Request-Type~~	~~25~~	~~6.3.25~~	~~Enumerated~~	~~M, V~~				~~No~~
User-Data-Already-Available	26	6.3.26	Enumerated	M, V				No
Confidentiality-Key	27	6.3.27	OctetString	M, V				No
Integrity-Key	28	6.3.28	OctetString	M, V				No
NOTE 1: The AVP header bit denoted as 'M', indicates whether support of the AVP is required. The AVP header bit denoted as 'V', indicates whether the optional Vendor-ID field is present in the AVP header. For further details, see IETF RFC 3588 [6]. NOTE 2: Depending on the concrete command.								

****** *Next Changed Section*******

6.3.25 ~~User-Data-Request-Type AVP~~Void

~~The User-Data-Request-Type AVP (AVP code 25) is of type Enumerated, and indicates the type of user profile the S-CSCF is requesting from the HSS. The following values are defined:~~

~~COMPLETE_PROFILE (0)~~

~~This value is used to request from the HSS the complete user profile corresponding to one or more public identities.~~

~~REGISTERED_PROFILE (1)~~

~~— This value is used to request from the HSS the registered part of the user profile corresponding to one or more public identities.~~

~~UNREGISTERED_PROFILE (2)~~

~~— This value is used to request from the HSS the unregistered part of the user profile corresponding to one or more public identities.~~Void

3GPP TSG CN WG4 Meeting #24 **N4-040966**
Sophia Antipolis, FRANCE, 16th – 20th AUGUST 2004

CR-Form-v7

CHANGE REQUEST

⌘ **29.229** **CR** **058** ⌘ **rev** **-** ⌘ Current version: **6.2.0** ⌘

*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 **X**

Title:	⌘	Simplification of the User Profile Split concept	
Source:	⌘	CN4	
Work item code:	⌘	IMS-CCR	***Date:*** ⌘ 23/06/2004
Category:	⌘	**A**	***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:
2 *(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)*

Reason for change: ⌘

This is an Essential Correction

Extensive discussion has taken place on the relative benefits and problems with the concept of splitting the user profile into Registered, Unregistered and Common parts. The user profile split was brought into the definition of the Cx interface at an early stage as a mechanism to allow for the otpimisation of the Cx interface in terms of
1. reducing the amount of data that is passed across it,
2. reducing the memory requirements placed on the S-CSCF, and
3. reducing the processing load in the S-CSCF.

However, because of the option to download parts of or the entire of the user profile, and because the S-CSCF should not have any concept of which part of the profile it actually has, with further consideration it has become clear that a number of significant problems exist.

- when performing a STORE_SERVER_NAME de-registration type which results in the subscriber being in the unregistered state but the S-CSCF has only the registered part of the User Profile, and has no mechanism to retrieve the unregistered part of the user profile.
- When a profile push occurs, there is only a requirement to send the part (ie the registered part or the unregistered part) of the user profile to the S-CSCF. However, if the S-CSCF currently holds the entire profile, the S-CSCF is obliged to overwrite that information with the part it receives and hence loses some subscriber information.
- It is currently not specified how the S-CSCF knows which parts of the user profile belong to the registered part or unregistered part of the service profile.
- It is not defined which values of the User-Data-Request-Type AVP are allowed in the cases when it's value is relevant; it is not e.g. sensible for the

	S-CSCF to ask for the unregistered profile part if the user is registered. Because of the increasingly complex solutions that are needed to fully solve all of these problems and maintain the concept of the user profile split, CN4 has become stalled in its discussions. For that reason, a completely alternative solution is considered: - to remove the option to download parts of the user profile altogether, and so only allow the complete profile to be downloaded, and - to add an indication to the IFC indicating the part to which it belongs. With this alternative solution the problems identified above are no longer in existance.
Summary of change: ⌘	The concept of downloading parts of the user profile is removed from the specs. An explicit identification of whether an IFC is related to the registered or unregistered subscriber state is added, allowing S-CSCF's to optimise their processing cycles upon receipt of SIP messages
Consequences if not approved: ⌘	The problems identified in 'Reasons for Change' continue. This leads to continuing problems from misalignment between the User Profile part stored at the S-CSCF and the user state, resulting in SIP messages being incorrectly processed and hence, subscribers experiencing incorrect behaviour from the network.

Clauses affected:	⌘	6.1.3, 6.3, 6.3.25			
Other specs affected:	⌘	**Y** / **N** **X** / ☐ Other core specifications ☐ / **X** Test specifications ☐ / **X** O&M Specifications	⌘	29.228 CR 124	
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.

6.1.3 Server-Assignment-Request (SAR) Command

The Server-Assignment-Request (SAR) command, indicated by the Command-Code field set to 301 and the 'R' bit set in the Command Flags field, is sent by a Diameter Multimedia client to a Diameter Multimedia server in order to request it to store the name of the server that is currently serving the user.

Message Format

<Server-Assignment-Request> ::= < Diameter Header: 301, 167772151, REQ, PXY >
< Session-Id >
{ Vendor-Specific-Application-Id }
{ Auth-Session-State }
{ Origin-Host }
{ Origin-Realm }
[Destination-Host]
{ Destination-Realm }
[User-Name]
***[Public-Identity]**
{ Server-Name }
{ Server-Assignment-Type }
~~**{ User-Data-Request-Type }**~~
{ User-Data-Already-Available }
*[AVP]
*[Proxy-Info]
*[Route-Record]

****** *Next Changed Section* ******

6.3 AVPs

The following table describes the Diameter AVPs defined for the Cx interface protocol, their AVP Code values, types, possible flag values and whether or not the AVP may be encrypted. The Vendor-Id header of all AVPs defined in this specification shall be set to 3GPP (10415).

Table 6.3.1: Diameter Multimedia Application AVPs

				AVP Flag rules				
Attribute Name	**AVP Code**	**Section defined**	**Value Type**	**Must**	**May**	**Should not**	**Must not**	**May Encr.**
Visited-Network-Identifier	1	6.3.1	OctetString	M, V				No
Public-Identity	2	6.3.2	UTF8String	M, V				N
Server-Name	3	6.3.3	UTF8String	M,V				No
Server-Capabilities	4	6.3.4	Grouped	M, V				No
Mandatory-Capability	5	6.3.5	Unsigned32	M, V				No
Optional-Capability	6	6.3.6	Unsigned32	M, V				No
User-Data	7	6.3.7	OctetString	M, V				No
SIP-Number-Auth-Items	8	6.3.8	Unsigned32	M, V				No
SIP-Authentication-Scheme	9	6.3.9	UTF8String	M, V				No

SIP-Authenticate	10	6.3.10	OctetString	M, V				No
SIP-Authorization	11	6.3.11	OctetString	M, V				No
SIP-Authentication-Context	12	6.3.12	OctetString	M, V				No
SIP-Auth-Data-Item	13	6.3.13	Grouped	M, V				No
SIP-Item-Number	14	6.3.14	Unsigned32	M, V				No
Server-Assignment-Type	15	6.3.15	Enumerated	M, V				No
Deregistration-Reason	16	6.3.16	Grouped	M, V				No
Reason-Code	17	6.3.17	Enumerated	M, V				No
Reason-Info	18	6.3.18	UTF8String	M, V				No
Charging-Information	19	6.3.19	Grouped	M, V				No
Primary-Event-Charging-Function-Name	20	6.3.20	DiameterURI	M, V				No
Secondary-Event-Charging-Function-Name	21	6.3.21	DiameterURI	M, V				No
Primary-Charging-Collection-Function-Name	22	6.3.22	DiameterURI	M, V				No
Secondary-Charging-Collection-Function-Name	23	6.3.23	DiameterURI	M, V				No
User-Authorization-Type	24	6.3.24	Enumerated	M, V				No
~~User-Data-Request-Type~~	~~25~~	~~6.3.25~~	~~Enumerated~~	~~M, V~~				~~No~~
User-Data-Already-Available	26	6.3.26	Enumerated	M, V				No
Confidentiality-Key	27	6.3.27	OctetString	M, V				No
Integrity-Key	28	6.3.28	OctetString	M, V				No
NOTE 1: The AVP header bit denoted as 'M', indicates whether support of the AVP is required. The AVP header bit denoted as 'V', indicates whether the optional Vendor-ID field is present in the AVP header. For further details, see IETF RFC 3588 [6]. NOTE 2: Depending on the concrete command.								

****** *Next Changed Section*******

6.3.25 ~~User-Data-Request-Type AVP~~Void

~~The User-Data-Request-Type AVP (AVP code 25) is of type Enumerated, and indicates the type of user profile the S-CSCF is requesting from the HSS. The following values are defined:~~

~~COMPLETE_PROFILE (0)~~

~~This value is used to request from the HSS the complete user profile corresponding to one or more public identities.~~

~~REGISTERED_PROFILE (1)~~

~~— This value is used to request from the HSS the registered part of the user profile corresponding to one or more public identities.~~

~~UNREGISTERED_PROFILE (2)~~

~~— This value is used to request from the HSS the unregistered part of the user profile corresponding to one or more public identities.~~Void

3GPP TSG CN WG4 Meeting #24 **N4-041161**
Sophia Antipolis, FRANCE, 16th – 20th AUGUST 2004

CR-Form-v7

CHANGE REQUEST

⌘ **29.228** **CR** **123** ⌘**rev** **2** ⌘ Current version: **5.8.0** ⌘

*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 **X**

Title:	⌘	Simplification of the User Profile Split concept		
Source:	⌘	CN4		
Work item code:	⌘	IMS-CCR	***Date:*** ⌘	23/06/2004
Category:	⌘	**F**	***Release:*** ⌘	Rel-5

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:
2 *(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)*

Reason for change: ⌘

This is an Essential Correction

Extensive discussion has taken place on the relative benefits and problems with the concept of splitting the user profile into Registered, Unregistered and Common parts. The user profile split was brought into the definition of the Cx interface at an early stage as a mechanism to allow for the otpimisation of the Cx interface in terms of
1. reducing the amount of data that is passed across it,
2. reducing the memory requirements placed on the S-CSCF, and
3. reducing the processing load in the S-CSCF.

However, because of the option to download parts of or the entire of the user profile, and because the S-CSCF should not have any concept of which part of the profile it actually has, with further consideration it has become clear that a number of significant problems exist.

- when performing a STORE_SERVER_NAME de-registration type which results in the subscriber being in the unregistered state but the S-CSCF has only the registered part of the User Profile, and has no mechanism to retrieve the unregistered part of the user profile.
- When a profile push occurs, there is only a requirement to send the part (ie the registered part or the unregistered part) of the user profile to the S-CSCF. However, if the S-CSCF currently holds the entire profile, the S-CSCF is obliged to overwrite that information with the part it receives and hence loses some subscriber information.
- It is currently not specified how the S-CSCF knows which parts of the user profile belong to the registered part or unregistered part of the service profile.
- It is not defined which values of the User-Data-Request-Type AVP are allowed in the cases when it's value is relevant; it is not e.g. sensible for the

	S-CSCF to ask for the unregistered profile part if the user is registered. Because of the increasingly complex solutions that are needed to fully solve all of these problems and maintain the concept of the user profile split, CN4 has become stalled in its discussions. For that reason, a completely alternative solution is considered: - to remove the option to download parts of the user profile altogether, and so only allow the complete profile to be downloaded, and - to add an indication to the IFC indicating the part to which it belongs. With this alternative solution the problems identified above are no longer in existance.
Summary of change: ⌘	The concept of downloading parts of the user profile is removed from the specs. An explicit identification of whether an IFC is related to the registered or unregistered subscriber state is added, allowing S-CSCF's to optimise their processing cycles upon receipt of SIP messages
Consequences if not approved: ⌘	The problems identified in 'Reasons for Change' continue. This leads to continuing problems from misalignment between the User Profile part stored at the S-CSCF and the user state, resulting in SIP messages being incorrectly processed and hence, subscribers experiencing incorrect behaviour from the network.

Clauses affected:	⌘	3.1, 6.1.2, 6.6, 6.6.1, 6.6.2, 7.15, Annex B.2.2, Annex E, CxDataSchema.xsd			
Other specs affected:	⌘	**Y** / **N** **X** /	Other core specifications	⌘	29.229 CR 057
		/ **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.

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply.

Common Part (of a user profile): Contains Initial Filter Criteria instances that should be evaluated both for registered and unregistered Public User Identities in the S-CSCF.

Complete user profile: Contains the Initial Filter Criteria instances of all three different user profile parts; registered part, unregistered part and common part.

IP Multimedia session: IP Multimedia session and IP Multimedia call are treated as equivalent in this specification.

Charging information: Data that is sent in the Charging-Information AVP.

Implicitly registered Public User identity set: A set of Public User Identities, which are registered and de-registered simultaneously when any of the Public User Identities belonging to that set is registered or de-registered.

Not Registered State: User is not Registered and has no S-CSCF assigned.

Registered Part (of a user profile): Contains Initial Filter Criteria instances that should be evaluated only for registered Public User Identities in the S-CSCF. iFCs from the registered part need not be evaluated when the user is unregistered.

Registered State: User is Registered at the request of the user and has an S-CSCF assigned.

Unregistered part (of a user profile): Contains Initial Filter Criteria instances that should be evaluated only for unregistered Public User Identities in the S-CSCF. iFCs from the unregistered part need not be evaluated when the user is registered.

Unregistered State: User is not Registered but has a serving S-CSCF assigned to execute Unregistered state services as a consequence of a terminating call or there is an S-CSCF keeping the user profile stored.

User information: The user related data that the S-CSCF requests from the HSS or HSS pushes to the S-CSCF, e.g. user profile and charging information.

User profile: Data that is sent in the User-Data AVP.

****** *Next Changed Section* ******

6.1.2 S-CSCF registration/deregistration notification

This procedure is used between the S-CSCF and the HSS. The procedure is invoked by the S-CSCF, corresponds to the combination of the operations Cx-Put and Cx-Pull (see 3GPP TS 23.228 [1]) and is used:

- To assign an S-CSCF to a Public User Identity, or to clear the name of the S-CSCF assigned to one or more Public User Identities.
- To download from HSS the relevant user information that the S-CSCF needs to serve the user.

This procedure is mapped to the commands Server-Assignment-Request/Answer in the Diameter application specified in 3GPP TS 29.229 [5]. Tables 6.1.2.1 and 6.1.2.2 describe the involved information elements.

Table 6.1.2.1: S-CSCF registration/deregistration notification request

Information element name	Mapping to Diameter AVP	Cat.	Description

Public User Identity (See 7.2)	Public-Identity	C	Public User Identity or list of Public User Identities. One and only one Public User Identity shall be present if the Server-Assignment-Type is any value other than TIMEOUT_DEREGISTRATION, USER_DEREGISTRATION or ADMINISTRATIVE_DEREGISTRATION. If Server-Assignment-Type indicates deregistration of some type and Private User Identityis not present in the request, at least one Public User Identity shall be present.
S-CSCF Name (See 7.4)	Server-Name	M	Name of the S-CSCF.
Private User Identity (See 7.3)	User-Name	C	Private User Identity. It shall be present if it is available when the S-CSCF issues the request. It may be absent during the initiation of a session to an unregistered user. In such a situation, Server-Assignment-Type shall contain the value UNREGISTERED_USER. In case of de-registration, Server-Assignment-Type equal to TIMEOUT_DEREGISTRATION, USER_DEREGISTRATION or ADMINISTRATIVE_DEREGISTRATION, if no Public User Identity is present then the Private User Identity shall be present.
Server Assignment Type (See 7.8)	Server-Assignment-Type	M	Type of update the S-CSCF requests in the HSS (e.g: de-registration). See 3GPP TS 29.229 [5] for all the possible values.
~~User Data Request Type~~ ~~(See 7.15)~~	~~User Data Request Type~~	~~M~~	~~Parts of the user profile the S-CSCF requests from the HSS (e.g: complete profile). See 3GPP TS 29.229 [5] for all the possible values.~~
User Data Already Available (See 7.16)	User-Data-Already-Available	M	This indicates if the user profile is already available in the S-CSCF.
Routing Information (See 7.13)	Destination-Host	C	If the S-CSCF knows the HSS name, the Destination-Host AVP shall be present in the command. This information is available if the request belongs to an already existing registration, e.g. in case of the re-registration, where the HSS name is stored in the S-CSCF. The HSS name is obtained from the Origin-Host AVP, which is received from the HSS, e.g. included in the MAA command. This information may not be available if the command is sent as a consequence of a session termination for an unregistered user. In this case the Destination-Host AVP is not present and the command is routed to the next Diameter node, e.g. SLF, based on the Diameter routing table in the S-CSCF.

Table 6.1.2.2: S-CSCF registration/deregistration notification response

Information element name	Mapping to Diameter AVP	Cat.	Description

Private User Identity (See 7.3)	User-Name	C	Private User identity. It shall be present if it is available when the HSS sends the response. It may be absent in the following error case: when the Server-Assignment-Type of the request is UNREGISTERED_USER and the received Public User Identity is not known by the HSS.
Registration result (See 7.6)	Result-Code / Experimental-Result	M	Result of registration. Result-Code AVP shall be used for errors defined in the Diameter Base Protocol. Experimental-Result AVP shall be used for Cx/Dx errors. This is a grouped AVP which contains the 3GPP Vendor ID in the Vendor-Id AVP, and the error code in the Experimental-Result-Code AVP.
User Profile (See 7.7)	User-Data	C	Relevant user profile. It shall be present when Server-Assignment-Type in the request is equal to NO_ASSIGNMENT, REGISTRATION, RE_REGISTRATION or UNREGISTERED_USER according to the rules defined in section 6.6. If the S-CSCF receives more data than it is prepared to accept, it shall perform the de-registration of the user with User-Authorization-Type set to DEREGISTRATION_TOO_MUCH_DATA and send back a SIP 3xx or 480 (Temporarily Unavailable) response, which shall trigger the selection of a new S-CSCF by the I-CSCF, as specified in 3GPP TS 24.229 [8].
Charging Information (See 7.12)	Charging-Information	C	Addresses of the charging functions. It shall be present when the User-Data AVP is sent to the S-CSCF. When this parameter is included, the Primary Charging Collection Function name shall be included. All other elements shall be included if they are available.

6.1.2.1 Detailed behaviour

On registering/deregistering a Public User Identity the S-CSCF shall inform the HSS. The same procedure is used by the S-CSCF to get the user profile. The relevant user profile downloaded is described in more detailed in sections 6.5.1 and 6.6. The HSS holds information about the state of registration of all the identities of the user. The S-CSCF uses this procedure to update such states. For implicitly registered identities, the rules defined in Section 6.5.1 shall apply.. The HSS shall, in the following order (in case of an error in any of the steps the HSS shall stop processing and return the corresponding error code, see 3GPP TS 29.229 [5]):

1. Check that the user is known. If not Experimental-Result-Code shall be set to DIAMETER_ERROR_USER_UNKNOWN.

2. The HSS may check whether the private and Public User Identities received in the request belong to the same user. If not Experimental-Result-Code shall be set to DIAMETER_ERROR_IDENTITIES_DONT_MATCH.

3. Check the Server Assignment Type value received in the request:

   - If it indicates REGISTRATION or RE_REGISTRATION, the HSS shall download the relevant user information. If set, the flag that indicates that the identity is pending of the confirmation of the authentication shall be cleared. The Result-Code shall be set to DIAMETER_SUCCESS and the HSS shall set the registration state of the Public User Identity as registered (if not already registered).

     Only one Public User Identity shall be present in the request. If more than one identity is present the Result-Code shall be set to DIAMETER_AVP_OCCURS_TOO_MANY_TIMES and no user information shall be returned.

   - If it indicates UNREGISTERED_USER, the HSS shall store the S-CSCF name, set the registration state of the Public User Identity as unregistered, i.e. registered as a consequence of a terminating call and download the relevant user information. The Result-Code shall be set to DIAMETER_SUCCESS.

Only one Public User Identity shall be present in the request. If more than one identity is present the Result-Code shall be set to DIAMETER_AVP_OCCURS_TOO_MANY_TIMES and the modifications specified in the previous paragraph shall not be performed.

- If it indicates TIMEOUT_DEREGISTRATION, USER_DEREGISTRATION, DEREGISTRATION_TOO_MUCH_DATA or ADMINISTRATIVE_DEREGISTRATION, the HSS shall clear the S-CSCF name for all the Public User Identities that the S-CSCF indicated in the request and set the registration state of the identities as not registered. If no public identity is present in the request, the Private User Identity shall be present; in this case the HSS shall clear the S-CSCF name for all the identities of the user and set their registration state to not registered. The Result-Code shall be set to DIAMETER_SUCCESS.

- If it indicates TIMEOUT_DEREGISTRATION_STORE_SERVER_NAME or USER_DEREGISTRATION_STORE_SERVER_NAME the HSS decides whether to keep the S-CSCF name stored or not for all the Public User Identities that the S-CSCF indicated in the request ~~and set the registration state of the identities as unregistered~~. If no Public User Identity is present in the request, the Private User Identity shall be present. If the HSS decided to keep the S-CSCF name stored, the HSS shall keep the S-CSCF name stored for all the identities of the user and set their registration state to unregistered. ~~If the S-CSCF has only the Registered part of the user profile stored it shall not indicate TIMEOUT_DEREGISTRATION_STORE_SERVER_NAME or USER_DEREGISTRATION_STORE_SERVER_NAME to the HSS.~~

  ~~If the HSS decides to keep the S-CSCF name t~~The Result-Code shall be set to DIAMETER_SUCCESS.

  If the HSS decides not to keep the S-CSCF name the Experimental-Result-Code shall be set to DIAMETER_SUCCESS_SERVER_NAME_NOT_STORED. If the HSS received Public User Identities in the request, the HSS shall set the registration state to not registered for the Public User Identity(ies) that the S-CSCF indicated in the request. If the HSS received a Private User Identity in the request, the HSS shall set the registration state of all Public User Identities related to the Private User Identity to not registered.

- If it indicates NO_ASSIGNMENT, the HSS checks whether the user is assigned for the S-CSCF requesting the data and download the ~~user information requested in the User-Data-Request-Type~~ relevant u~~U~~ser information. The Result-Code shall be set to DIAMETER_SUCCESS. If the requesting S-CSCF is not the same as the assigned S-CSCF, the Result-Code shall be set to DIAMETER_UNABLE_TO COMPLY.

  Only one Public User Identity shall be present in the request. If more than one Public User Identity is present the Result-Code shall be set to DIAMETER_AVP_OCCURS_TOO_MANY_TIMES and no user information shall be returned.

- If it indicates AUTHENTICATION_FAILURE or AUTHENTICATION_TIMEOUT, the HSS shall clear the S-CSCF name for the Public User Identity that the S-CSCF indicated in the request and set the registration state of the identity as not registered. The flag that indicates that the identity is pending confirmation of the authentication shall be cleared. The Result-Code shall be set to DIAMETER_SUCCESS.

  Only one Public User Identity shall be present in the request. If more than one identity is present the Result-Code shall be set to DIAMETER_AVP_OCCURS_TOO_MANY_TIMES and the modifications specified in the previous paragraph shall not be performed.

If the HSS cannot fulfil the received request, e.g. due to database error, it shall set the Result-Code to DIAMETER_UNABLE_TO_COMPLY. The HSS shall not modify any user state nor download any user information to the S-CSCF.

See chapter 8.1.2 and 8.1.3 for the description of the handling of the error situations: reception of an S-CSCF name different from the one stored in the HSS and reception of a Server-Assignment-Type value not compatible with the registration state of the user.

******* Next Changed Section*******

6.6 Download of the Relevant User Profile

The download of the relevant profile data from the HSS to the S-CSCF depends on whether the user profile is already stored in the S-CSCF ~~and/or on the user profile requested from the S-CSCF and/or whether the requested user profile is up-to-date in the S-CSCF~~.

If User-Data-Already-Available is set to USER_DATA_NOT_AVAILABLE the HSS shall download the requested user profile. If the Public User Identity in the request is included in an i~~I~~mplicitly r~~R~~egistered Public User Identity s~~S~~et, the HSS shall include in the response the service~~user~~ profiles associated with all ~~other~~ Public User I~~i~~dentities within the~~at~~ i~~I~~mplicitly r~~R~~egistered Public User Identity s~~S~~et to which the received Public User Identity belongs. ~~, according to the value of User-Data-Request-Type. See Section 6.3.25 in 3GPP TS 29.229 [5].~~

~~If User-Data-Already-Available is set to USER_DATA_ALREADY_AVAILABLE and the requested profile is not up-to-date (according to the indications stored in HSS defined in 6.6.1) the HSS shall download the requested profile, according to the value of User-Data-Request-Type. See Section 6.3.25 in 3GPP TS 29.229 [5].~~

If User-Data-Already-Available is set to USER_DATA_ALREADY_AVAILABLE~~Otherwise~~, the HSS shall not return any user profile data.

6.6.1 HSS initiated update of User Profile

The request to update of the user information in the S-CSCF includes only the Public User Identities of the implicitly registered Public User identity set with the associated service profiles. See 6.5.2.1.

If the Public User Identity is registered or unregistered (i.e. registered as a consequence of a terminating call or there is a S-CSCF keeping the user profile stored) and there are changes in ~~the registered part of~~ the user profile, the HSS shall immediately push the complete user profile to the S-CSCF ~~the registered part of the user profile~~.

~~If the Public User Identity is registered and there are changes in the unregistered part of the user profile, the HSS shall set a flag indicating that the unregistered part of the profile is not up-to-date in the S-CSCF. The HSS shall not initiate any push toward the S-CSCF.~~

~~If the Public User Identity is unregistered (i.e. registered as a consequence of a terminating call or there is a S-CSCF keeping the user profile stored) and there is a change in the unregistered part of the user profile, the HSS shall immediately push to the S-CSCF the unregistered part of the user profile.~~

~~If the Public User Identity is unregistered (i.e. registered as a consequence of a terminating call or there is a S-CSCF keeping the user profile stored) and there is a change in the registered part of the user profile, the HSS shall set a flag indicating that the registered part of the profile is not up-to-date in the S-CSCF. The HSS shall not initiate any push toward the S-CSCF.~~

6.6.2 S-CSCF operation

At deregistration of a Public User Identity, ~~T~~the S-CSCF shall store the user information if it sends Server-Assignment-Request command including Server-Assignment-Type AVP set to value USER_DEREGISTRATION_STORE_SERVER_NAME or TIMEOUT_DEREGISTRATION_STORE_SERVER_NAME and the HSS responds with DIAMETER_SUCCESS. Otherwise the S-CSCF shall not keep user information.

****** *Next Changed Section* ******

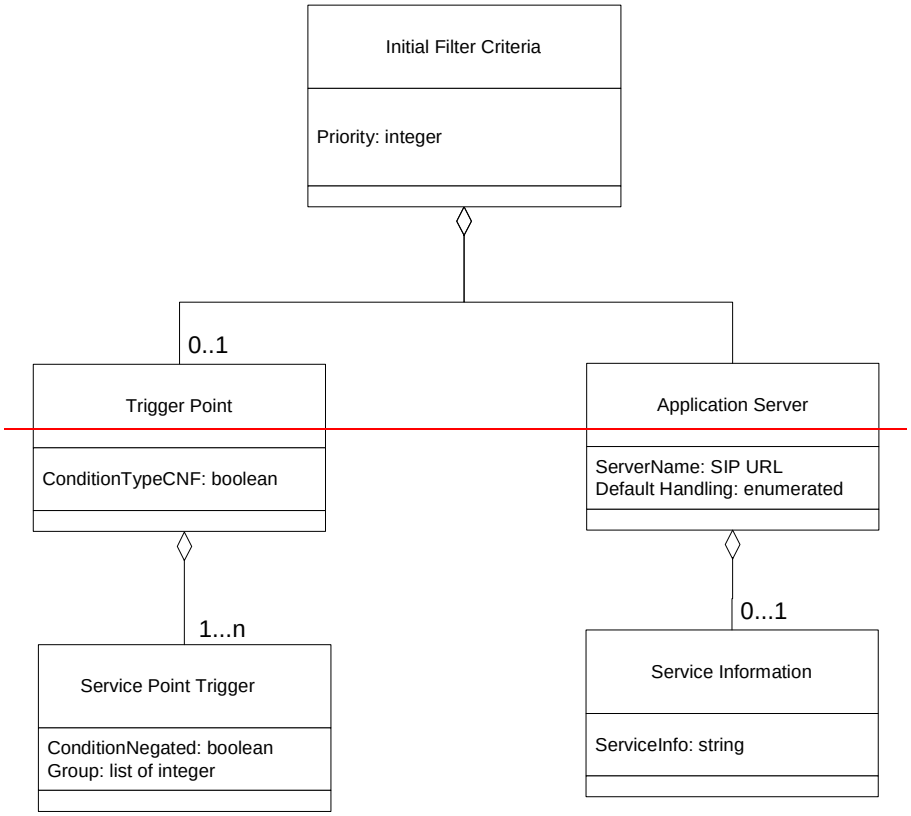
7.15 ~~User-Data-Request-Type~~Void

~~Part of the user profile the S-CSCF requests from the HSS. See 3GPP TS 29. 229 [5] for a list of values.~~Void

****** *Next Changed Section* ******

B.2.2 Initial Filter Criteria

The following picture gives an outline of the UML model of Initial Filter Criteria class:



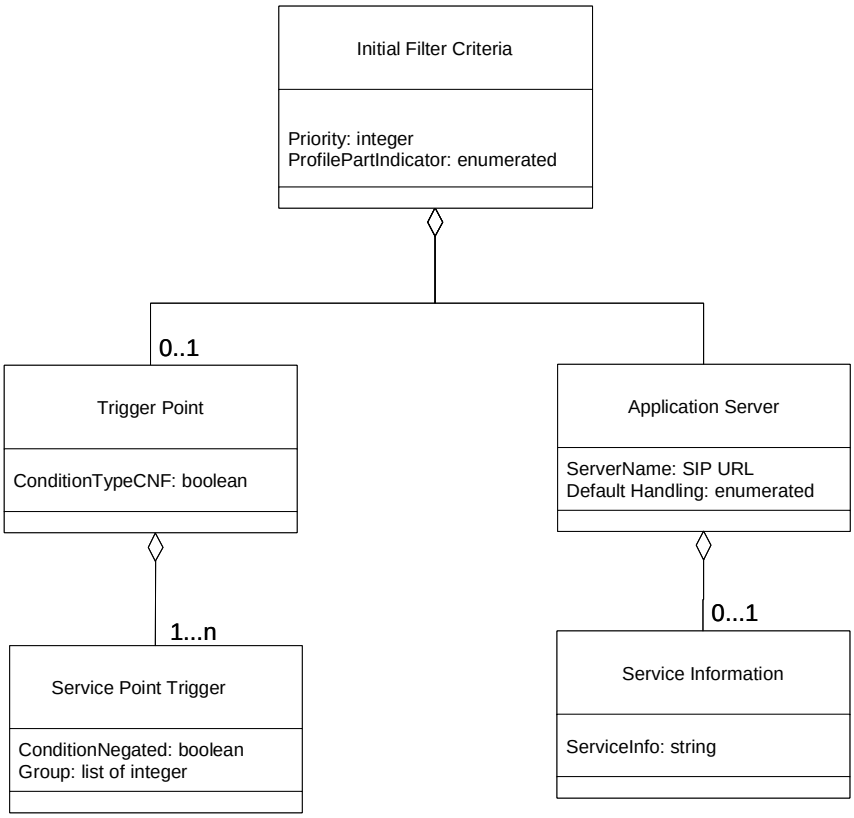


Figure B.2.2.1.1: Initial Filter Criteria

Each instance of the Initial Filter Criteria class is composed of zero or one instance of a Trigger Point class and one instance of an Application Server class. Priority indicates the priority of the Filter Criteria. The higher the Priority Number the lower the priority of the Filter Criteria is; i.e., a Filter Criteria with a higher value of Priority Number shall be assessed after the Filter Criteria with a smaller Priority Number have been assessed. The same priority shall not be assigned to more than one initial Filter Criterion. ~~ConditionTypeCNF is a boolean that is TRUE when the Trigger Point associated with the FilterCriteria is a boolean expresion in Conjuctive Normal Form (CNF) and FALSE if the Trigger Point is expressed in Disjunctive Normal Form (DNF) (see Annex C).~~

ProfilePartIndicator attribute ~~represents~~is an enumerated type, with possible values "~~Registered"~~ REGISTERED and~~or~~ UNREGISTERED~~"Unregistered"~~, indicating if the i~~I~~FC is a part of the r~~R~~egistered or u~~U~~nregistered u~~U~~ser p~~P~~rofile. If ProfilePartIndicator is missing from the i~~I~~FC, the i~~I~~FC is considered to be relevant to both the r~~R~~egistered and u~~U~~nregistered parts of the u~~U~~ser p~~P~~rofile, i.e. belongs to the c~~C~~ommon part of the u~~U~~ser p~~P~~rofile.

Trigger Point class describes the trigger points that should be checked in order to find out if the indicated Application Server should be contacted or not. Each TriggerPoint is a boolean expression in Conjuctive or Disjunctive Normal form (CNF of DNF). The absence of Trigger Point instance will indicate an unconditional triggering to Application Server.

The attribute ConditionTypeCNF attribute defines how the set of SPTs are expressed, i.e. either an Ored set of ANDed sets of SPT statements or an ANDed set of Ored sets of statements. Individual SPTstatements can also be negated. These combinations are termed, respectively, Disjunctive Normal Form (DNF) and Conjunctive Normal Form (CNF) for the SPT (see Annex C). Both DNF and CNF forms can be used. ConditionTypeCNF is a boolean that is TRUE when the Trigger Point associated with the FilterCriteria is a boolean expresion in Conjuctive Normal Form (CNF) and FALSE if the Trigger Point is expressed in Disjunctive Normal Form (DNF) (see Annex C).

Each Trigger Point is composed by 1 to n instances of the class Service Point Trigger.

Application Server class defines the application server, which is contacted, if the trigger points are met. Server Name is the SIP URL of the application server to contact. Default Handling determines whether the dialog should be released if the Application Server could not be reached or not; it is of type enumerated and can take the values: SESSION_CONTINUED or SESSION_TERMINATED.

The Application Server class contains zero or one instance of the Service Information class. Service Information class allows to download to S-CSCF information that is to be transferred transparently to an Application Server when the

trigger points of a filter criterion are satisfied. ServiceInformation is a string conveying that information. See 3GPP TS 23.218 [7] for a description of the use of this information element.

****** *Next Changed Section*******

Annex E (normative): XML schema for the Cx interface user profile

The file CxDataType.xsd, attached to this specification, contains the XML schema for the Cx interface user profile. Such XML schema details all the data types on which XML documents containing Cx profile information shall be based. The XML schema file is intended to be used by an XML parser.

Table E.1 describes the data types and the dependencies among them that configure the XML schema.

Table E.1: XML schema for Cx interface: simple data types

Data type	Tag	Base type	Comments
tPriority	Priority	integer	>= 0
tProfilePartIndicator	ProfilePartIndicator	enumerated	Possible values: 0 (REGISTERED) 1 (UNREGISTERED)
tGroupID	Group	integer	>= 0
tDefaultHandling	DefaultHandling	enumerated	Possible values: 0 (SESSION_CONTINUED) 1 (SESSION_TERMINATED)
tDirectionOfRequest	SessionCase	enumerated	Possible values: 0 (ORIGINATING_SESSION) 1 TERMINATING_ REGISTERED 2 (TERMINATING_UNREGISTERED)
tPrivateID	PrivateID	anyURI	Syntax described in RFC 2486
tSIP_URL	Identity	anyURI	Syntax described in RFC 3261
tTEL_URL	Identity	anyURI	Syntax described in RFC 2806
tIdentity	Identity	(union)	Union of tSIP_URL and tTEL_URL
tServiceInfo	ServiceInfo	string	
tString	RequestURI, Method, Header, Content, Line	string	
tBool	ConditionTypeCNF, ConditionNegated, BarringIndication	boolean	Possible values: 0 (false) 1 (true)
tSubscribedMediaProfileId	SubscribedMediaProfileId	integer	>=0

Table E.2: XML schema for Cx interface: complex data types

Data type	Tag	Compound of		
		Tag	Type	Cardinality
tIMSSubscription	IMSSubscription	PrivateID	tPrivateID	1
		ServiceProfile	tServiceProfile	(1 to n)
tServiceProfile	ServiceProfile	PublicIdentity	tPublicIdentity	(1 to n)
		InitialFilterCriteria	tInitialFilterCriteria	(0 to n)
		CoreNetworkServicesAuthorization	CoreNetworkServicesAuthorization	(0 to 1)
tCoreNetworkServicesAuthorization	CoreNetworkServicesAuthorization	SubscribedMediaProfileId	tSubscribedMediaProfileId	(0 to 1)
tPublicIdentity	PublicIdentity	BarringIndication	tBool	1
		Identity	tIdentity	1
tInitialFilterCriteria	InitialFilterCriteria	Priority	tPriority	1
		TriggerPoint	tTrigger	(0 to 1)
		ApplicationServer	tApplicationServer	1
		ProfilePartIndicator	tProfilePartIndicator	(0 to 1)
tTrigger	TriggerPoint	ConditionTypeCNF	tBool	1
		SPT	tSePoTri	(1 to n)
tSePoTri	SPT	ConditionNegated	tBool	(0 to 1)
		Group	tGroupID	(1 to n)
		Choice of: RequestURI	tString	1
		Choice of: Method	tString	1
		Choice of: SIPHeader	tHeader	1
		Choice of: SessionCase	tDirectionOfRequest	1
		Choice of: SessionDescription	tSessionDescription	1
tHeader	SIPHeader	Header	tString	1

		Content	tString	(0 to 1)
tSessionDescription	SessionDescription	Line	tString	1
		Content	tString	(0 to 1)
tApplicationServer	ApplicationServer	ServerName	tSIP_URL	1
		DefaultHandling	tDefaultHandling	(0 to 1)
		ServiceInfo	tServiceInfo	(0 to 1)
NOTE: “n” shall be interpreted as non-bounded.				

****** Changes to .xsd file ******

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:simpleType name="tPriority" final="list restriction">
        <xs:restriction base="xs:int">
            <xs:minInclusive value="0"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tProfilePartIndicator" final="list restriction">
        <xs:restriction base="xs:unsignedByte">
            <xs:maxInclusive value="1"/>
            <xs:enumeration value="0">
                <xs:annotation>
                    <xs:documentation>
                        <label xml:lang="en">REGISTERED</label>
                        <definition xml:lang="en">iFC is pPart of the rRegistered pProfile</definition>
                    </xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="1">
                <xs:annotation>
                    <xs:documentation>
                        <label xml:lang="en">UNREGISTERED</label>
                        <definition xml:lang="en">iFC is pPart of the uUnregistered pProfile</definition>
                    </xs:documentation>
                </xs:annotation>
            </xs:enumeration>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tGroupID" final="list restriction">
        <xs:restriction base="xs:int">
            <xs:minInclusive value="0"/>
        </xs:restriction>
```

```
</xs:simpleType>
<xs:simpleType name="tDefaultHandling" final="list restriction">
    <xs:restriction base="xs:unsignedByte">
        <xs:maxInclusive value="1"/>
        <xs:enumeration value="0">
            <xs:annotation>
                <xs:documentation>
                    <label xml:lang="en">SESSION_CONTINUED</label>
                    <definition xml:lang="en">Session Continued</definition>
                </xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="1">
            <xs:annotation>
                <xs:documentation>
                    <label xml:lang="en">SESSION_TERMINATED</label>
                    <definition xml:lang="en">Session Terminated</definition>
                </xs:documentation>
            </xs:annotation>
        </xs:enumeration>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tDirectionOfRequest" final="list restriction">
    <xs:restriction base="xs:unsignedByte">
        <xs:maxInclusive value="3"/>
        <xs:enumeration value="0">
            <xs:annotation>
                <xs:documentation>
                    <label xml:lang="en">ORIGINATING_SESSION</label>
                    <definition xml:lang="en">Originating Session</definition>
                </xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="1">
            <xs:annotation>
```

```
                <xs:documentation>
                    <label xml:lang="en">TERMINATING_REGISTERED</label>
                    <definition xml:lang="en">Terminating Session for registered user</definition>
                </xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="2">
            <xs:annotation>
                <xs:documentation>
                    <label xml:lang="en">TERMINATING_UNREGISTERED</label>
                    <definition xml:lang="en">Terminating Session for unregistered user</definition>
                </xs:documentation>
            </xs:annotation>
        </xs:enumeration>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tPrivateID" final="list restriction">
    <xs:restriction base="xs:anyURI"/>
</xs:simpleType>
<xs:simpleType name="tSIP_URL" final="list restriction">
    <xs:restriction base="xs:anyURI"/>
</xs:simpleType>
<xs:simpleType name="tTEL_URL" final="list restriction">
    <xs:restriction base="xs:anyURI"/>
</xs:simpleType>
<xs:simpleType name="tIdentity" final="list restriction">
    <xs:union memberTypes="tSIP_URL tTEL_URL"/>
</xs:simpleType>
<xs:simpleType name="tServiceInfo" final="list restriction">
    <xs:restriction base="xs:string">
        <xs:minLength value="0"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tString" final="list restriction">
    <xs:restriction base="xs:string">
```

```
            <xs:minLength value="0"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tBool">
        <xs:restriction base="xs:boolean"/>
    </xs:simpleType>
    <xs:simpleType name="tSubscribedMediaProfileId" final="list restriction">
        <xs:restriction base="xs:int">
            <xs:minInclusive value="0"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="tIMSSubscription">
        <xs:sequence>
            <xs:element name="PrivateID" type="tPrivateID"/>
            <xs:element name="ServiceProfile" type="tServiceProfile" maxOccurs="unbounded"/>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tServiceProfile">
        <xs:sequence>
            <xs:element name="PublicIdentity" type="tPublicIdentity" maxOccurs="unbounded"/>
            <xs:element name="CoreNetworkServicesAuthorization" type="tCoreNetworkServicesAuthorization"
minOccurs="0"/>
            <xs:element name="InitialFilterCriteria" type="tInitialFilterCriteria" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tCoreNetworkServicesAuthorization">
        <xs:sequence>
            <xs:element name="SubscribedMediaProfileId" type="tSubscribedMediaProfileId" minOccurs="0"/>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tInitialFilterCriteria">
        <xs:sequence>
```

```
            <xs:element name="Priority" type="tPriority"/>
            <xs:element name="TriggerPoint" type="tTrigger" minOccurs="0"/>
            <xs:element name="ApplicationServer" type="tApplicationServer"/>
            <xs:element name="ProfilePartIndicator" type="tProfilePartIndicator" minOccurs="0"/>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tTrigger">
        <xs:sequence>
            <xs:element name="ConditionTypeCNF" type="tBool"/>
            <xs:element name="SPT" type="tSePoTri" maxOccurs="unbounded"/>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tSePoTri">
        <xs:sequence>
            <xs:element name="ConditionNegated" type="tBool" default="0" minOccurs="0"/>
            <xs:element name="Group" type="tGroupID" maxOccurs="unbounded"/>
            <xs:choice>
                <xs:element name="RequestURI" type="tString"/>
                <xs:element name="Method" type="tString"/>
                <xs:element name="SIPHeader" type="tHeader"/>
                <xs:element name="SessionCase" type="tDirectionOfRequest"/>
                <xs:element name="SessionDescription" type="tSessionDescription"/>
            </xs:choice>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tHeader">
        <xs:sequence>
            <xs:element name="Header" type="tString"/>
            <xs:element name="Content" type="tString" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tSessionDescription">
```

```
        <xs:sequence>
            <xs:element name="Line" type="tString"/>
            <xs:element name="Content" type="tString" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tApplicationServer">
        <xs:sequence>
            <xs:element name="ServerName" type="tSIP_URL"/>
            <xs:element name="DefaultHandling" type="tDefaultHandling" minOccurs="0"/>
            <xs:element name="ServiceInfo" type="tServiceInfo" minOccurs="0"/>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tPublicIdentity">
        <xs:sequence>
            <xs:element name="BarringIndication" type="tBool" default="0" minOccurs="0"/>
            <xs:element name="Identity" type="tIdentity"/>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="IMSSubscription" type="tIMSSubscription"/>
</xs:schema>
```

3GPP TSG CN WG4 Meeting #24 **N4-041162**
Sophia Antipolis, FRANCE, 16th – 20th AUGUST 2004

CR-Form-v7

CHANGE REQUEST

⌘ **29.228** **CR** **124** ⌘ **rev** **2** ⌘ Current version: **6.3.0** ⌘

*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 **X**

Title:	⌘	Simplification of the User Profile Split concept		
Source:	⌘	CN4		
Work item code:	⌘	IMS-CCR	***Date:*** ⌘	23/06/2004
Category:	⌘	**A**	***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:
2 *(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)*

Reason for change: ⌘

This is an Essential Correction

Extensive discussion has taken place on the relative benefits and problems with the concept of splitting the user profile into Registered, Unregistered and Common parts. The user profile split was brought into the definition of the Cx interface at an early stage as a mechanism to allow for the otpimisation of the Cx interface in terms of
1. reducing the amount of data that is passed across it,
2. reducing the memory requirements placed on the S-CSCF, and
3. reducing the processing load in the S-CSCF.

However, because of the option to download parts of or the entire of the user profile, and because the S-CSCF should not have any concept of which part of the profile it actually has, with further consideration it has become clear that a number of significant problems exist.

- when performing a STORE_SERVER_NAME de-registration type which results in the subscriber being in the unregistered state but the S-CSCF has only the registered part of the User Profile, and has no mechanism to retrieve the unregistered part of the user profile.
- When a profile push occurs, there is only a requirement to send the part (ie the registered part or the unregistered part) of the user profile to the S-CSCF. However, if the S-CSCF currently holds the entire profile, the S-CSCF is obliged to overwrite that information with the part it receives and hence loses some subscriber information.
- It is currently not specified how the S-CSCF knows which parts of the user profile belong to the registered part or unregistered part of the service profile.
- It is not defined which values of the User-Data-Request-Type AVP are allowed in the cases when it's value is relevant; it is not e.g. sensible for the

	S-CSCF to ask for the unregistered profile part if the user is registered. Because of the increasingly complex solutions that are needed to fully solve all of these problems and maintain the concept of the user profile split, CN4 has become stalled in its discussions. For that reason, a completely alternative solution is considered: - to remove the option to download parts of the user profile altogether, and so only allow the complete profile to be downloaded, and - to add an indication to the IFC indicating the part to which it belongs. With this alternative solution the problems identified above are no longer in existance.
Summary of change: ⌘	The concept of downloading parts of the user profile is removed from the specs. An explicit identification of whether an IFC is related to the registered or unregistered subscriber state is added, allowing S-CSCF's to optimise their processing cycles upon receipt of SIP messages
Consequences if not approved: ⌘	The problems identified in 'Reasons for Change' continue. This leads to continuing problems from misalignment between the User Profile part stored at the S-CSCF and the user state, resulting in SIP messages being incorrectly processed and hence, subscribers experiencing incorrect behaviour from the network.

Clauses affected:	⌘	3.1, 6.1.2, 6.6, 6.6.1, 6.6.2, 7.15, Annex B.2.2, Annex E, CxDataSchema.xsd			
Other specs affected:	⌘	**Y** / **N** **X** /	Other core specifications	⌘	29.229 CR 058
		/ **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.

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply.

Common Part (of a user profile): Contains Initial Filter Criteria instances that should be evaluated both for registered and unregistered Public User Identities in the S-CSCF.

Complete user profile: Contains the Initial Filter Criteria instances of all three different user profile parts; registered part, unregistered part and common part.

IP Multimedia session: IP Multimedia session and IP Multimedia call are treated as equivalent in this specification.

Charging information: Data that is sent in the Charging-Information AVP.

Implicitly registered Public User identity set: A set of Public User Identities, which are registered and de-registered simultaneously when any of the Public User Identities belonging to that set is registered or de-registered.

Not Registered State: User is not Registered and has no S-CSCF assigned.

Registered Part (of a user profile): Contains Initial Filter Criteria instances that should be evaluated only for registered Public User Identities in the S-CSCF. iFCs from the registered part need not be evaluated when the user is unregistered.

Registered State: User is Registered at the request of the user and has an S-CSCF assigned.

Unregistered part (of a user profile): Contains Initial Filter Criteria instances that should be evaluated only for unregistered Public User Identities in the S-CSCF. iFCs from the unregistered part need not be evaluated when the user is registered.

Unregistered State: User is not Registered but has a serving S-CSCF assigned to execute Unregistered state services as a consequence of a terminating call or there is an S-CSCF keeping the user profile stored.

User information: The user related data that the S-CSCF requests from the HSS or HSS pushes to the S-CSCF, e.g. user profile and charging information.

User profile: Data that is sent in the User-Data AVP.

****** *Next Changed Section* ******

6.1.2 S-CSCF registration/deregistration notification

This procedure is used between the S-CSCF and the HSS. The procedure is invoked by the S-CSCF, corresponds to the combination of the operations Cx-Put and Cx-Pull (see 3GPP TS 23.228 [1]) and is used:

- To assign an S-CSCF to a Public User Identity, or to clear the name of the S-CSCF assigned to one or more Public User Identities.
- To download from HSS the relevant user information that the S-CSCF needs to serve the user.

This procedure is mapped to the commands Server-Assignment-Request/Answer in the Diameter application specified in 3GPP TS 29.229 [5]. Tables 6.1.2.1 and 6.1.2.2 describe the involved information elements.

Table 6.1.2.1: S-CSCF registration/deregistration notification request

Information element name	Mapping to Diameter AVP	Cat.	Description

Public User Identity (See 7.2)	Public-Identity	C	Public User Identity or list of Public User Identities. One and only one Public User Identity shall be present if the Server-Assignment-Type is any value other than TIMEOUT_DEREGISTRATION, USER_DEREGISTRATION or ADMINISTRATIVE_DEREGISTRATION. If Server-Assignment-Type indicates deregistration of some type and Private User Identityis not present in the request, at least one Public User Identity shall be present.
S-CSCF Name (See 7.4)	Server-Name	M	Name of the S-CSCF.
Private User Identity (See 7.3)	User-Name	C	Private User Identity. It shall be present if it is available when the S-CSCF issues the request. It may be absent during the initiation of a session to an unregistered user. In such a situation, Server-Assignment-Type shall contain the value UNREGISTERED_USER. In case of de-registration, Server-Assignment-Type equal to TIMEOUT_DEREGISTRATION, USER_DEREGISTRATION or ADMINISTRATIVE_DEREGISTRATION, if no Public User Identity is present then the Private User Identity shall be present.
Server Assignment Type (See 7.8)	Server-Assignment-Type	M	Type of update the S-CSCF requests in the HSS (e.g: de-registration). See 3GPP TS 29.229 [5] for all the possible values.
~~User Data Request Type~~ ~~(See 7.15)~~	~~User Data Request Type~~	~~M~~	~~Parts of the user profile the S-CSCF requests from the HSS (e.g: complete profile). See 3GPP TS 29.229 [5] for all the possible values.~~
User Data Already Available (See 7.16)	User-Data-Already-Available	M	This indicates if the user profile is already available in the S-CSCF.
Routing Information (See 7.13)	Destination-Host	C	If the S-CSCF knows the HSS name, the Destination-Host AVP shall be present in the command. This information is available if the request belongs to an already existing registration, e.g. in case of the re-registration, where the HSS name is stored in the S-CSCF. The HSS name is obtained from the Origin-Host AVP, which is received from the HSS, e.g. included in the MAA command. This information may not be available if the command is sent as a consequence of a session termination for an unregistered user. In this case the Destination-Host AVP is not present and the command is routed to the next Diameter node, e.g. SLF, based on the Diameter routing table in the S-CSCF.

Table 6.1.2.2: S-CSCF registration/deregistration notification response

Information element name	Mapping to Diameter AVP	Cat.	Description

Private User Identity (See 7.3)	User-Name	C	Private User identity. It shall be present if it is available when the HSS sends the response. It may be absent in the following error case: when the Server-Assignment-Type of the request is UNREGISTERED_USER and the received Public User Identity is not known by the HSS.
Registration result (See 7.6)	Result-Code / Experimental-Result	M	Result of registration. Result-Code AVP shall be used for errors defined in the Diameter Base Protocol. Experimental-Result AVP shall be used for Cx/Dx errors. This is a grouped AVP which contains the 3GPP Vendor ID in the Vendor-Id AVP, and the error code in the Experimental-Result-Code AVP.
User Profile (See 7.7)	User-Data	C	Relevant user profile. It shall be present when Server-Assignment-Type in the request is equal to NO_ASSIGNMENT, REGISTRATION, RE_REGISTRATION or UNREGISTERED_USER according to the rules defined in section 6.6. If the S-CSCF receives more data than it is prepared to accept, it shall perform the de-registration of the user with User-Authorization-Type set to DEREGISTRATION_TOO_MUCH_DATA and send back a SIP 3xx or 480 (Temporarily Unavailable) response, which shall trigger the selection of a new S-CSCF by the I-CSCF, as specified in 3GPP TS 24.229 [8].
Charging Information (See 7.12)	Charging-Information	C	Addresses of the charging functions. It shall be present when the User-Data AVP is sent to the S-CSCF. When this parameter is included, the Primary Charging Collection Function name shall be included. All other elements shall be included if they are available.

6.1.2.1 Detailed behaviour

On registering/deregistering a Public User Identity the S-CSCF shall inform the HSS. The same procedure is used by the S-CSCF to get the user profile. The relevant user profile downloaded is described in more detailed in sections 6.5.1 and 6.6. The HSS holds information about the state of registration of all the identities of the user. The S-CSCF uses this procedure to update such states. For implicitly registered identities, the rules defined in Section 6.5.1 shall apply.. The HSS shall, in the following order (in case of an error in any of the steps the HSS shall stop processing and return the corresponding error code, see 3GPP TS 29.229 [5]):

1. Check that the user is known. If not Experimental-Result-Code shall be set to DIAMETER_ERROR_USER_UNKNOWN.

2. The HSS may check whether the private and Public User Identities received in the request belong to the same user. If not Experimental-Result-Code shall be set to DIAMETER_ERROR_IDENTITIES_DONT_MATCH.

3. Check the Server Assignment Type value received in the request:

   - If it indicates REGISTRATION or RE_REGISTRATION, the HSS shall download the relevant user information. If set, the flag that indicates that the identity is pending of the confirmation of the authentication shall be cleared. The Result-Code shall be set to DIAMETER_SUCCESS and the HSS shall set the registration state of the Public User Identity as registered (if not already registered).

     Only one Public User Identity shall be present in the request. If more than one identity is present the Result-Code shall be set to DIAMETER_AVP_OCCURS_TOO_MANY_TIMES and no user information shall be returned.

   - If it indicates UNREGISTERED_USER, the HSS shall store the S-CSCF name, set the registration state of the Public User Identity as unregistered, i.e. registered as a consequence of a terminating call and download the relevant user information. The Result-Code shall be set to DIAMETER_SUCCESS.

Only one Public User Identity shall be present in the request. If more than one identity is present the Result-Code shall be set to DIAMETER_AVP_OCCURS_TOO_MANY_TIMES and the modifications specified in the previous paragraph shall not be performed.

- If it indicates TIMEOUT_DEREGISTRATION, USER_DEREGISTRATION, DEREGISTRATION_TOO_MUCH_DATA or ADMINISTRATIVE_DEREGISTRATION, the HSS shall clear the S-CSCF name for all the Public User Identities that the S-CSCF indicated in the request and set the registration state of the identities as not registered. If no public identity is present in the request, the Private User Identity shall be present; in this case the HSS shall clear the S-CSCF name for all the identities of the user and set their registration state to not registered. The Result-Code shall be set to DIAMETER_SUCCESS.

- If it indicates TIMEOUT_DEREGISTRATION_STORE_SERVER_NAME or USER_DEREGISTRATION_STORE_SERVER_NAME the HSS decides whether to keep the S-CSCF name stored or not for all the Public User Identities that the S-CSCF indicated in the ~~request and set the registration state of the identities as unregistered~~. If no Public User Identity is present in the request, the Private User Identity shall be present. If the HSS decided to keep the S-CSCF name stored, the HSS shall keep the S-CSCF name stored for all the identities of the user and set their registration state to unregistered. ~~If the S-CSCF has only the Registered part of the user profile stored it shall not indicate TIMEOUT_DEREGISTRATION_STORE_SERVER_NAME or USER_DEREGISTRATION_STORE_SERVER_NAME to the HSS.~~

  ~~If the HSS decides to keep the S-CSCF name t~~The Result-Code shall be set to DIAMETER_SUCCESS.

  If the HSS decides not to keep the S-CSCF name the Experimental-Result-Code shall be set to DIAMETER_SUCCESS_SERVER_NAME_NOT_STORED. If the HSS received Public User Identities in the request, the HSS shall set the registration state to not registered for the Public User Identity(ies) that the S-CSCF indicated in the request. If the HSS received a Private User Identity in the request, the HSS shall set the registration state of all Public User Identities related to the Private User Identity to not registered.

- If it indicates NO_ASSIGNMENT, the HSS checks whether the user is assigned for the S-CSCF requesting the data and download the ~~user information requested in the User-Data-Request-Type~~ relevant u~~U~~ser information. The Result-Code shall be set to DIAMETER_SUCCESS. If the requesting S-CSCF is not the same as the assigned S-CSCF, the Result-Code shall be set to DIAMETER_UNABLE_TO COMPLY.

  Only one Public User Identity shall be present in the request. If more than one Public User Identity is present the Result-Code shall be set to DIAMETER_AVP_OCCURS_TOO_MANY_TIMES and no user information shall be returned.

- If it indicates AUTHENTICATION_FAILURE or AUTHENTICATION_TIMEOUT, the HSS shall clear the S-CSCF name for the Public User Identity that the S-CSCF indicated in the request and set the registration state of the identity as not registered. The flag that indicates that the identity is pending confirmation of the authentication shall be cleared. The Result-Code shall be set to DIAMETER_SUCCESS.

  Only one Public User Identity shall be present in the request. If more than one identity is present the Result-Code shall be set to DIAMETER_AVP_OCCURS_TOO_MANY_TIMES and the modifications specified in the previous paragraph shall not be performed.

If the HSS cannot fulfil the received request, e.g. due to database error, it shall set the Result-Code to DIAMETER_UNABLE_TO_COMPLY. The HSS shall not modify any user state nor download any user information to the S-CSCF.

See chapter 8.1.2 and 8.1.3 for the description of the handling of the error situations: reception of an S-CSCF name different from the one stored in the HSS and reception of a Server-Assignment-Type value not compatible with the registration state of the user.

******* Next Changed Section*******

6.6 Download of the Relevant User Profile

The download of the relevant profile data from the HSS to the S-CSCF depends on whether the user profile is already stored in the S-CSCF ~~and/or on the user profile requested from the S-CSCF and/or whether the requested user profile is up-to-date in the S-CSCF~~.

If User-Data-Already-Available is set to USER_DATA_NOT_AVAILABLE the HSS shall download the requested user profile. If the Public User Identity in the request is included in an i~~I~~mplicitly r~~R~~egistered Public User Identity s~~S~~et, the HSS shall include in the response the service~~user~~ profiles associated with all ~~other~~ Public User I~~i~~dentities within the~~at~~ i~~I~~mplicitly r~~R~~egistered Public User Identity s~~S~~et to which the received Public User Identity belongs. ~~, according to the value of User-Data-Request-Type. See Section 6.3.25 in 3GPP TS 29.229 [5].~~

~~If User-Data-Already-Available is set to USER_DATA_ALREADY_AVAILABLE and the requested profile is not up-to-date (according to the indications stored in HSS defined in 6.6.1) the HSS shall download the requested profile, according to the value of User-Data-Request-Type. See Section 6.3.25 in 3GPP TS 29.229 [5].~~

If User-Data-Already-Available is set to USER_DATA_ALREADY_AVAILABLE~~Otherwise~~, the HSS shall not return any user profile data.

6.6.1 HSS initiated update of User Profile

The request to update of the user information in the S-CSCF includes only the Public User Identities of the implicitly registered Public User identity set with the associated service profiles. See 6.5.2.1.

If the Public User Identity is registered or unregistered (i.e. registered as a consequence of a terminating call or there is a S-CSCF keeping the user profile stored) and there are changes in ~~the registered part of~~ the user profile, the HSS shall immediately push the complete user profile to the S-CSCF ~~the registered part of the user profile~~.

~~If the Public User Identity is registered and there are changes in the unregistered part of the user profile, the HSS shall set a flag indicating that the unregistered part of the profile is not up-to-date in the S-CSCF. The HSS shall not initiate any push toward the S-CSCF.~~

~~If the Public User Identity is unregistered (i.e. registered as a consequence of a terminating call or there is a S-CSCF keeping the user profile stored) and there is a change in the unregistered part of the user profile, the HSS shall immediately push to the S-CSCF the unregistered part of the user profile.~~

~~If the Public User Identity is unregistered (i.e. registered as a consequence of a terminating call or there is a S-CSCF keeping the user profile stored) and there is a change in the registered part of the user profile, the HSS shall set a flag indicating that the registered part of the profile is not up-to-date in the S-CSCF. The HSS shall not initiate any push toward the S-CSCF.~~

6.6.2 S-CSCF operation

At deregistration of a Public User Identity, ~~T~~the S-CSCF shall store the user information if it sends Server-Assignment-Request command including Server-Assignment-Type AVP set to value USER_DEREGISTRATION_STORE_SERVER_NAME or TIMEOUT_DEREGISTRATION_STORE_SERVER_NAME and the HSS responds with DIAMETER_SUCCESS. Otherwise the S-CSCF shall not keep user information.

****** *Next Changed Section* ******

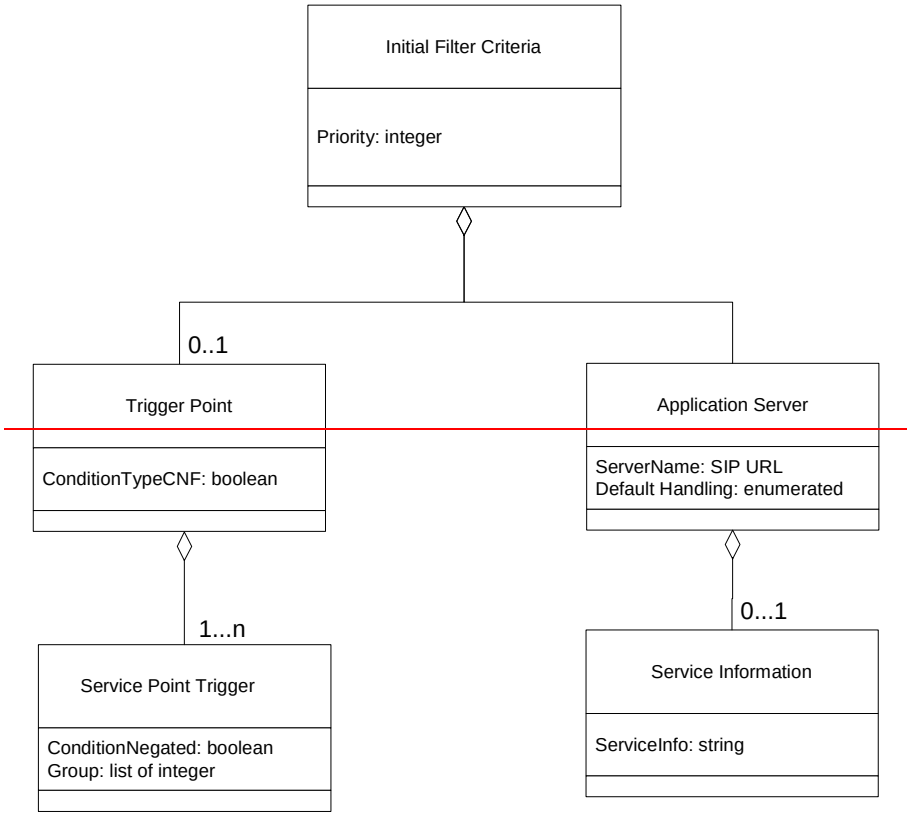
7.15 ~~User-Data-Request-Type~~Void

~~Part of the user profile the S-CSCF requests from the HSS. See 3GPP TS 29. 229 [5] for a list of values.~~Void

****** *Next Changed Section* ******

B.2.2 Initial Filter Criteria

The following picture gives an outline of the UML model of Initial Filter Criteria class:



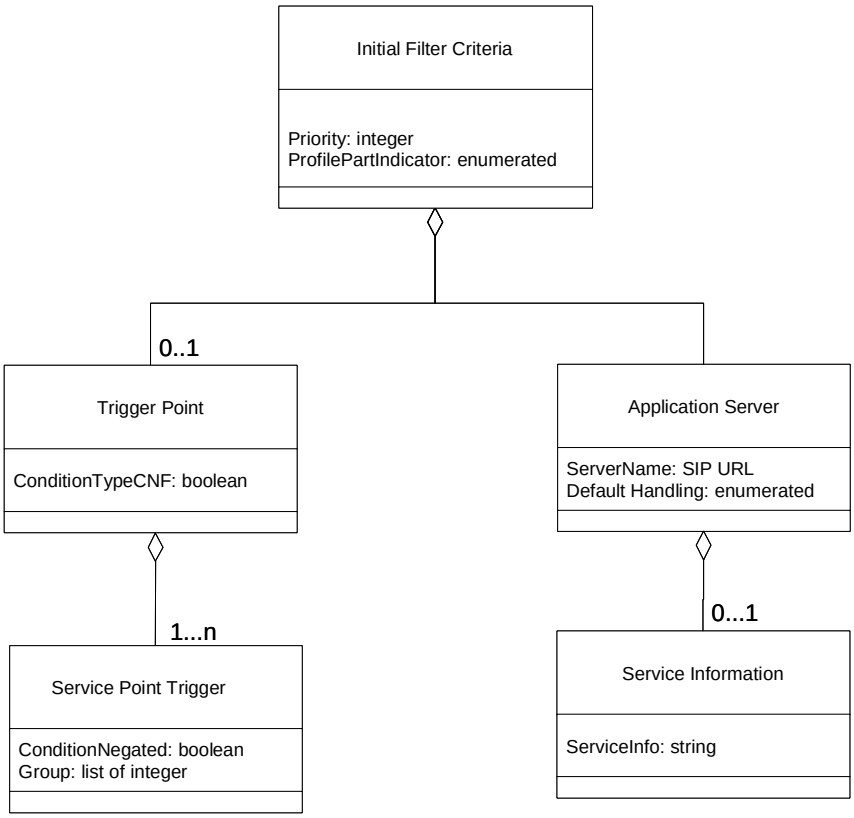


Figure B.2.2.1.1: Initial Filter Criteria

Each instance of the Initial Filter Criteria class is composed of zero or one instance of a Trigger Point class and one instance of an Application Server class. Priority indicates the priority of the Filter Criteria. The higher the Priority Number the lower the priority of the Filter Criteria is; i.e., a Filter Criteria with a higher value of Priority Number shall be assessed after the Filter Criteria with a smaller Priority Number have been assessed. The same priority shall not be assigned to more than one initial Filter Criterion. ~~ConditionTypeCNF is a boolean that is TRUE when the Trigger Point associated with the FilterCriteria is a boolean expresion in Conjuctive Normal Form (CNF) and FALSE if the Trigger Point is expressed in Disjunctive Normal Form (DNF) (see Annex C).~~

ProfilePartIndicator attribute ~~represents~~is an enumerated type, with possible values "~~Registered"~~ REGISTERED and~~or~~ UNREGISTERED~~"Unregistered"~~, indicating if the i~~I~~FC is a part of the r~~R~~egistered or u~~U~~nregistered u~~U~~ser p~~P~~rofile. If ProfilePartIndicator is missing from the i~~I~~FC, the i~~I~~FC is considered to be relevant to both the r~~R~~egistered and u~~U~~nregistered parts of the u~~U~~ser p~~P~~rofile, i.e. belongs to the c~~C~~ommon part of the u~~U~~ser p~~P~~rofile.

Trigger Point class describes the trigger points that should be checked in order to find out if the indicated Application Server should be contacted or not. Each TriggerPoint is a boolean expression in Conjuctive or Disjunctive Normal form (CNF of DNF). The absence of Trigger Point instance will indicate an unconditional triggering to Application Server.

The attribute ConditionTypeCNF attribute defines how the set of SPTs are expressed, i.e. either an Ored set of ANDed sets of SPT statements or an ANDed set of Ored sets of statements. Individual SPTstatements can also be negated. These combinations are termed, respectively, Disjunctive Normal Form (DNF) and Conjunctive Normal Form (CNF) for the SPT (see Annex C). Both DNF and CNF forms can be used. ConditionTypeCNF is a boolean that is TRUE when the Trigger Point associated with the FilterCriteria is a boolean expresion in Conjuctive Normal Form (CNF) and FALSE if the Trigger Point is expressed in Disjunctive Normal Form (DNF) (see Annex C).

Each Trigger Point is composed by 1 to n instances of the class Service Point Trigger.

Application Server class defines the application server, which is contacted, if the trigger points are met. Server Name is the SIP URL of the application server to contact. Default Handling determines whether the dialog should be released if the Application Server could not be reached or not; it is of type enumerated and can take the values: SESSION_CONTINUED or SESSION_TERMINATED.

The Application Server class contains zero or one instance of the Service Information class. Service Information class allows to download to S-CSCF information that is to be transferred transparently to an Application Server when the

trigger points of a filter criterion are satisfied. ServiceInformation is a string conveying that information. See 3GPP TS 23.218 [7] for a description of the use of this information element.

****** *Next Changed Section*******

Annex E (normative): XML schema for the Cx interface user profile

The file CxDataType.xsd, attached to this specification, contains the XML schema for the Cx interface user profile. Such XML schema details all the data types on which XML documents containing Cx profile information shall be based. The XML schema file is intended to be used by an XML parser.

Table E.1 describes the data types and the dependencies among them that configure the XML schema.

Table E.1: XML schema for Cx interface: simple data types

Data type	Tag	Base type	Comments
tPriority	Priority	integer	>= 0
tProfilePartIndicator	ProfilePartIndicator	enumerated	Possible values: 0 (REGISTERED) 1 (UNREGISTERED)
tGroupID	Group	integer	>= 0
tDefaultHandling	DefaultHandling	enumerated	Possible values: 0 (SESSION_CONTINUED) 1 (SESSION_TERMINATED)
tDirectionOfRequest	SessionCase	enumerated	Possible values: 0 (ORIGINATING_SESSION) 1 TERMINATING_ REGISTERED 2 (TERMINATING_UNREGISTERED)
tPrivateID	PrivateID	anyURI	Syntax described in RFC 2486
tSIP_URL	Identity	anyURI	Syntax described in RFC 3261
tTEL_URL	Identity	anyURI	Syntax described in RFC 2806
tIdentity	Identity	(union)	Union of tSIP_URL and tTEL_URL
tServiceInfo	ServiceInfo	string	
tString	RequestURI, Method, Header, Content, Line	string	
tBool	ConditionTypeCNF, ConditionNegated, BarringIndication	boolean	Possible values: 0 (false) 1 (true)
tSubscribedMediaProfileId	SubscribedMediaProfileId	integer	>=0

Table E.2: XML schema for Cx interface: complex data types

Data type	Tag	Compound of		
		Tag	Type	Cardinality
tIMSSubscription	IMSSubscription	PrivateID	tPrivateID	1
		ServiceProfile	tServiceProfile	(1 to n)
tServiceProfile	ServiceProfile	PublicIdentity	tPublicIdentity	(1 to n)
		InitialFilterCriteria	tInitialFilterCriteria	(0 to n)
		CoreNetworkServicesAuthorization	CoreNetworkServicesAuthorization	(0 to 1)
tCoreNetworkServicesAuthorization	CoreNetworkServicesAuthorization	SubscribedMediaProfileId	tSubscribedMediaProfileId	(0 to 1)
tPublicIdentity	PublicIdentity	BarringIndication	tBool	1
		Identity	tIdentity	1
tInitialFilterCriteria	InitialFilterCriteria	Priority	tPriority	1
		TriggerPoint	tTrigger	(0 to 1)
		ApplicationServer	tApplicationServer	1
		ProfilePartIndicator	tProfilePartIndicator	(0 to 1)
tTrigger	TriggerPoint	ConditionTypeCNF	tBool	1
		SPT	tSePoTri	(1 to n)
tSePoTri	SPT	ConditionNegated	tBool	(0 to 1)
		Group	tGroupID	(1 to n)
		Choice of: RequestURI	tString	1
		Choice of: Method	tString	1
		Choice of: SIPHeader	tHeader	1
		Choice of: SessionCase	tDirectionOfRequest	1
		Choice of: SessionDescription	tSessionDescription	1
tHeader	SIPHeader	Header	tString	1

		Content	tString	(0 to 1)
tSessionDescription	SessionDescription	Line	tString	1
		Content	tString	(0 to 1)
tApplicationServer	ApplicationServer	ServerName	tSIP_URL	1
		DefaultHandling	tDefaultHandling	(0 to 1)
		ServiceInfo	tServiceInfo	(0 to 1)
NOTE: “n” shall be interpreted as non-bounded.				

****** Changes to .xsd file ******

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:simpleType name="tPriority" final="list restriction">
        <xs:restriction base="xs:int">
            <xs:minInclusive value="0"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tProfilePartIndicator" final="list restriction">
        <xs:restriction base="xs:unsignedByte">
            <xs:maxInclusive value="1"/>
            <xs:enumeration value="0">
                <xs:annotation>
                    <xs:documentation>
                        <label xml:lang="en">REGISTERED</label>
                        <definition xml:lang="en">iFC is pPart of the rRegistered pProfile</definition>
                    </xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="1">
                <xs:annotation>
                    <xs:documentation>
                        <label xml:lang="en">UNREGISTERED</label>
                        <definition xml:lang="en">iFC is pPart of the uUnregistered pProfile</definition>
                    </xs:documentation>
                </xs:annotation>
            </xs:enumeration>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tGroupID" final="list restriction">
        <xs:restriction base="xs:int">
            <xs:minInclusive value="0"/>
        </xs:restriction>
```

```
</xs:simpleType>
<xs:simpleType name="tDefaultHandling" final="list restriction">
    <xs:restriction base="xs:unsignedByte">
        <xs:maxInclusive value="1"/>
        <xs:enumeration value="0">
            <xs:annotation>
                <xs:documentation>
                    <label xml:lang="en">SESSION_CONTINUED</label>
                    <definition xml:lang="en">Session Continued</definition>
                </xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="1">
            <xs:annotation>
                <xs:documentation>
                    <label xml:lang="en">SESSION_TERMINATED</label>
                    <definition xml:lang="en">Session Terminated</definition>
                </xs:documentation>
            </xs:annotation>
        </xs:enumeration>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tDirectionOfRequest" final="list restriction">
    <xs:restriction base="xs:unsignedByte">
        <xs:maxInclusive value="3"/>
        <xs:enumeration value="0">
            <xs:annotation>
                <xs:documentation>
                    <label xml:lang="en">ORIGINATING_SESSION</label>
                    <definition xml:lang="en">Originating Session</definition>
                </xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="1">
            <xs:annotation>
```

```
                <xs:documentation>
                    <label xml:lang="en">TERMINATING_REGISTERED</label>
                    <definition xml:lang="en">Terminating Session for registered user</definition>
                </xs:documentation>
            </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="2">
            <xs:annotation>
                <xs:documentation>
                    <label xml:lang="en">TERMINATING_UNREGISTERED</label>
                    <definition xml:lang="en">Terminating Session for unregistered user</definition>
                </xs:documentation>
            </xs:annotation>
        </xs:enumeration>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tPrivateID" final="list restriction">
    <xs:restriction base="xs:anyURI"/>
</xs:simpleType>
<xs:simpleType name="tSIP_URL" final="list restriction">
    <xs:restriction base="xs:anyURI"/>
</xs:simpleType>
<xs:simpleType name="tTEL_URL" final="list restriction">
    <xs:restriction base="xs:anyURI"/>
</xs:simpleType>
<xs:simpleType name="tIdentity" final="list restriction">
    <xs:union memberTypes="tSIP_URL tTEL_URL"/>
</xs:simpleType>
<xs:simpleType name="tServiceInfo" final="list restriction">
    <xs:restriction base="xs:string">
        <xs:minLength value="0"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="tString" final="list restriction">
    <xs:restriction base="xs:string">
```

```
            <xs:minLength value="0"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="tBool">
        <xs:restriction base="xs:boolean"/>
    </xs:simpleType>
    <xs:simpleType name="tSubscribedMediaProfileId" final="list restriction">
        <xs:restriction base="xs:int">
            <xs:minInclusive value="0"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="tIMSSubscription">
        <xs:sequence>
            <xs:element name="PrivateID" type="tPrivateID"/>
            <xs:element name="ServiceProfile" type="tServiceProfile" maxOccurs="unbounded"/>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tServiceProfile">
        <xs:sequence>
            <xs:element name="PublicIdentity" type="tPublicIdentity" maxOccurs="unbounded"/>
            <xs:element name="CoreNetworkServicesAuthorization" type="tCoreNetworkServicesAuthorization"
minOccurs="0"/>
            <xs:element name="InitialFilterCriteria" type="tInitialFilterCriteria" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tCoreNetworkServicesAuthorization">
        <xs:sequence>
            <xs:element name="SubscribedMediaProfileId" type="tSubscribedMediaProfileId" minOccurs="0"/>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tInitialFilterCriteria">
        <xs:sequence>
```

```
            <xs:element name="Priority" type="tPriority"/>
            <xs:element name="TriggerPoint" type="tTrigger" minOccurs="0"/>
            <xs:element name="ApplicationServer" type="tApplicationServer"/>
            <xs:element name="ProfilePartIndicator" type="tProfilePartIndicator" minOccurs="0"/>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tTrigger">
        <xs:sequence>
            <xs:element name="ConditionTypeCNF" type="tBool"/>
            <xs:element name="SPT" type="tSePoTri" maxOccurs="unbounded"/>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tSePoTri">
        <xs:sequence>
            <xs:element name="ConditionNegated" type="tBool" default="0" minOccurs="0"/>
            <xs:element name="Group" type="tGroupID" maxOccurs="unbounded"/>
            <xs:choice>
                <xs:element name="RequestURI" type="tString"/>
                <xs:element name="Method" type="tString"/>
                <xs:element name="SIPHeader" type="tHeader"/>
                <xs:element name="SessionCase" type="tDirectionOfRequest"/>
                <xs:element name="SessionDescription" type="tSessionDescription"/>
            </xs:choice>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tHeader">
        <xs:sequence>
            <xs:element name="Header" type="tString"/>
            <xs:element name="Content" type="tString" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tSessionDescription">
```

```
        <xs:sequence>
            <xs:element name="Line" type="tString"/>
            <xs:element name="Content" type="tString" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tApplicationServer">
        <xs:sequence>
            <xs:element name="ServerName" type="tSIP_URL"/>
            <xs:element name="DefaultHandling" type="tDefaultHandling" minOccurs="0"/>
            <xs:element name="ServiceInfo" type="tServiceInfo" minOccurs="0"/>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="tPublicIdentity">
        <xs:sequence>
            <xs:element name="BarringIndication" type="tBool" default="0" minOccurs="0"/>
            <xs:element name="Identity" type="tIdentity"/>
            <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:element name="IMSSubscription" type="tIMSSubscription"/>
</xs:schema>
```