**3GPP TSG-SA5 Meeting #140-e *S5-216093***

**Online, , 15th Nov 2021 - 24th Nov 2021**

|  |
| --- |
| *CR-Form-v12.1* |
| **CHANGE REQUEST** |
|  |
|  | **28.541** | **CR** | **0614** | **rev** | 1 | **Current version:** | **17.4.0** |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network | **x** | Core Network |  |

|  |
| --- |
|  |
| ***Title:***  | Rel-17 CR 28.541 Stage 3 YANG updates for stage 2 CRs 214164, 214585-8 |
|  |  |
| ***Source to WG:*** | Cisco Systems Belgium |
| ***Source to TSG:*** | S5 |
|  |  |
| ***Work item code:*** | adNRM |  | ***Date:*** | 2021-11-04 |
|  |  |  |  |  |
| ***Category:*** | F |  | ***Release:*** | Rel-17 |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)…Rel-15 (Release 15)Rel-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)* |
|  |  |
| ***Reason for change:*** | The YANG mapping of some earlier agreed Stage 2 elements is not complete. It is fixed here. |
|  |  |
| ***Summary of change:*** | Only YANG is updated to match stage 2. |
|  |  |
| ***Consequences if not approved:*** | Faulty/Missing parts in the YANG solution set. |
|  |  |
| ***Clauses affected:*** | 5A.3.2-3, H.5.2, H.5.5, H.5.7, H.5.10, H.5.12, H.5.14, H.5.16-18, H.5.20-25, H.5.X |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **x** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **x** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **x** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** | Forge link: https://forge.3gpp.org/rep/sa5/MnS/tree/28.541\_Rel17\_CR\_0614\_stage3\_YANG\_updates/yang-models |
|  |  |
| ***This CR's revision history:*** |  |

|  |
| --- |
| **1st Change** |

### 5A.3.2 EP\_N70

#### 5A.3.2.1 Definition

This IOC represents the N70 interface between S/I-CSCF and HSS, which is defined in 3GPP TS 23.501 [2].

#### 5A.3.2.2 Attributes

The EP\_N70 IOC includes attributes inherited from EP\_RP IOC (defined in TS 28.622[30]) and the following attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| localAddress | O | T | T | F | T |
| remoteAddress | O | T | T | F | T |

NOTE: There is no YANG or YAML representation of the HSS, so this IOC cannot be accessed through those interfaces.

### 5A.3.3 EP\_N71

#### 5A.3.3.1 Definition

This IOC represents the N71 interface between AF and HSS, which is defined in 3GPP TS 23.501 [2].

#### 5A.3.3.2 Attributes

The EP\_N71 IOC includes attributes inherited from EP\_RP IOC (defined in TS 28.622[30]) and the following attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| localAddress | O | T | T | F | T |
| remoteAddress | O | T | T | F | T |

NOTE: There is no YANG or YAML representation of the HSS, so this IOC cannot be accessed through those interfaces.

|  |
| --- |
| **2nd Change** |

## H.5.2 module \_3gpp-5gc-nrm-amffunction.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-amffunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-amffunction;

 prefix amf3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import ietf-inet-types { prefix inet; }

 import \_3gpp-common-top { prefix top3gpp; }

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 import \_3gpp-nr-nrm-gnbcuupfunction { prefix gnbcuup3gpp; }

 organization "3gpp SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "AMFFunction derived from basic ManagedFunction.";

 reference "3GPP TS 28.541";

 revision 2021-11-01 { reference CR-0614 ; }

 revision 2020-11-05 { reference CR-0412 ; }

 revision 2019-10-25 { reference "S5-194457 S5-193518"; }

 revision 2019-05-31 { reference "Ericsson refactoring."; }

 revision 2018-08-07 { reference "Initial revision"; }

 grouping TaiRange {

 description "This <<dataType>> represents the range of TAIs.";

 uses types3gpp:PLMNId;

 list nRTACRangeList {

 description "The range of TACs.";

 key idx;

 leaf idx {

 description "Synthetic index for the element.";

 type uint32;

 }

 uses nRTACRange;

 }

 }

 grouping nRTACRange {

 choice start-end-or-pattern {

 description "Either the start and end attributes, or the pattern

 attribute, shall be present.";

 case start-end {

 leaf nRTACstart {

 description "First value identifying the start of a TAC range,

 to be used when the range of TAC's can be represented as a

 hexadecimal range (e.g., TAC ranges). 3-octet string identifying

 a tracking area code, each character in the string shall take a

 value of '0' to '9' or 'A' to 'F' and shall represent 4 bits.

 The most significant character representing the 4 most significant

 bits of the TAC shall appear first in the string, and the

 character representing the 4 least significant bit of the TAC

 shall appear last in the string.";

 type string {

 pattern "([A-Fa-f0-9]{4}|[A-Fa-f0-9]{6})";

 }

 mandatory true; // Start + end mandatory if this case chosen

 }

 leaf nRTACend {

 description "Last value identifying the end of a TAC range, to

 be used when the range of TAC's can be represented as a

 hexadecimal range (e.g. TAC ranges). 3-octet string

 identifying a tracking area code, each character in the

 string shall take a value of '0' to '9' or 'A' to 'F' and

 shall represent 4 bits. The most significant character

 representing the 4 most significant bits of the TAC shall

 appear first in the string, and the character representing

 the 4 least significant bit of the TAC shall appear last in

 the string.";

 type string {

 pattern "([A-Fa-f0-9]{4}|[A-Fa-f0-9]{6})";

 }

 mandatory true;

 }

 }

 case pattern {

 leaf nRTACpattern {

 description "Pattern (regular expression according to the

 ECMA-262 dialect [x0]) representing the set of TAC's

 belonging to this range. A TAC value is considered part of the

 range if and only if the TAC string fully matches the regular

 expression.";

 type string;

 }

 }

 mandatory true;

 }

 }

 grouping AMFFunctionGrp {

 description "Represents the AMFFunction IOC";

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNInfoList {

 min-elements 1;

 description "It defines the PLMN(s) of a Network Function. ";

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 list aMFIdentifier {

 description "An AMF identifier, comprising an AMF Region ID, an

 AMF Set ID and an AMF Pointer.";

 min-elements 1;

 max-elements 1;

 key "amfRegionId amfSetId amfPointer";

 uses types3gpp:AmfIdentifier;

 }

 leaf sBIFQDN {

 description "The FQDN of the registered NF instance in the

 service-based interface.";

 type inet:domain-name;

 }

 leaf interPlmnFQDN {

 description "If the NF needs to be discoverable by other NFs

 in a different PLMN, then an FQDN that is used for inter-PLMN

 routing as specified in 3GPP TS 23.003 [13] shall be registered

 with the NRF.";

 type inet:domain-name;

 }

 list taiList {

 description "The list of TAIs.";

 min-elements 1;

 key idx;

 leaf idx { type uint32; }

 uses gnbcuup3gpp:TAIGrp;

 }

 list taiRangeList {

 description "The range of TAIs.";

 min-elements 1;

 key idx;

 leaf idx { type uint32; }

 uses TaiRange;

 }

 leaf-list cNSIIdList {

 description "It is a set of NSI ID. NSI ID is an identifier

 for identifying the Core Network part of a Network Slice

 instance when multiple Network Slice instances of the same

 Network Slice are deployed, and there is a need to

 differentiate between them in the 5GC, see clause 3.1 of

 TS 23.501 [2] and subclause 6.1.6.2.7 of 3GPP TS 29.531 [24].";

 config false;

 type string;

 }

 list gUAMIdList {

 description "List of supported Globally Unique AMF Ids (GUAMIs).";

 config false;

 min-elements 1;

 key idx;

 leaf idx { type uint32; }

 uses nfp3gpp:Guami;

 }

 list managedNFProfile {

 description "Profile definition of a Managed NF";

 reference "TS 23.501";

 key idx;

 min-elements 1;

 max-elements 1;

 uses nfp3gpp:ManagedNFProfile;

 }

 list commModelList {

 min-elements 1;

 key "groupId";

 description "Specifies a list of commModel. It can be used by NF and

 NF services to interact with each other in 5G Core network ";

 reference "3GPP TS 23.501";

 uses types5g3gpp:CommModel;

 }

 leaf-list aMFSet {

 config false;

 description "The AMFSet that the AFM Function is associated with.";

 type instance-identifier;

 }

 }

 augment "/me3gpp:ManagedElement" {

 list AMFFunction {

 description "5G Core AMF Function";

 reference "3GPP TS 28.541";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses AMFFunctionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses;

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **3rd Change** |

## H.5.5 module \_3gpp-5gc-nrm-ausffunction.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-ausffunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-ausffunction;

 prefix ausf3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import ietf-inet-types { prefix inet; }

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import \_3gpp-common-top { prefix top3gpp; }

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 organization "3gpp SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "This IOC represents the AUSF function in 5GC. For more

 information about the AUSF, see 3GPP TS 23.501.";

 reference "3GPP TS 28.541";

 revision 2021-11-01 { reference Refactoring ; }

 revision 2020-11-05 { reference CR-0412 ; }

 revision 2019-10-25 { reference "S5-194457 S5-193518"; }

 revision 2019-05-22 {reference "initial revision"; }

 grouping AUSFFuntionGrp {

 description "Represents the AUSFFuntion IOC";

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNIdList {

 description "List of at most six entries of PLMN Identifiers, but at

 least one (the primary PLMN Id).

 The PLMN Identifier is composed of a Mobile Country Code (MCC) and

 a Mobile Network Code (MNC).";

 min-elements 1;

 max-elements 6;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 leaf sBIFQDN {

 description "The FQDN of the registered NF instance in the

 service-based interface.";

 type inet:domain-name;

 }

 list sNSSAIList {

 description "List of S-NSSAIs the managed object is capable of supporting.

 (Single Network Slice Selection Assistance Information)

 An S-NSSAI has an SST (Slice/Service type) and an optional SD

 (Slice Differentiator) field.";

 //optional support

 reference "3GPP TS 23.003";

 key "sd sst";

 uses types5g3gpp:SNssai;

 }

 list managedNFProfile {

 description "Profile definition of a Managed NF";

 key idx;

 min-elements 1;

 max-elements 1;

 uses nfp3gpp:ManagedNFProfile;

 }

 list commModelList {

 min-elements 1;

 key "groupId";

 description "Specifies a list of commModel. It can be used by NF and

 NF services to interact with each other in 5G Core network ";

 reference "3GPP TS 23.501";

 uses types5g3gpp:CommModel;

 }

 }

 augment "/me3gpp:ManagedElement" {

 list AUSFFunction {

 description "5G Core AUSF Function";

 reference "3GPP TS 28.541";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses AUSFFuntionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses;

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **4th Change** |

## H.5.7 module \_3gpp-5gc-nrm-ep.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-ep {

 yang-version 1.1;

 namespace 'urn:3gpp:sa5:\_3gpp-5gc-nrm-ep';

 prefix "cep3gpp";

 import \_3gpp-common-ep-rp { prefix eprp3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import \_3gpp-5gc-nrm-affunction { prefix af3gpp; }

 import \_3gpp-5gc-nrm-amffunction { prefix amf3gpp; }

 import \_3gpp-5gc-nrm-ausffunction { prefix ausf3gpp; }

 import \_3gpp-5gc-nrm-dnfunction { prefix dn3gpp; }

 import \_3gpp-5gc-nrm-lmffunction { prefix lmf3gpp; }

 import \_3gpp-5gc-nrm-n3iwffunction { prefix n3iwf3gpp; }

 import \_3gpp-5gc-nrm-ngeirfunction { prefix ngeir3gpp; }

 import \_3gpp-5gc-nrm-nrffunction { prefix nrf3gpp; }

 import \_3gpp-5gc-nrm-nssffunction { prefix nssf3gpp; }

 import \_3gpp-5gc-nrm-pcffunction { prefix pcf3gpp; }

 import \_3gpp-5gc-nrm-seppfunction { prefix sepp3gpp; }

 import \_3gpp-5gc-nrm-smffunction { prefix smf3gpp; }

 import \_3gpp-5gc-nrm-smsffunction { prefix smsf3gpp; }

 import \_3gpp-5gc-nrm-udmfunction { prefix udm3gpp; }

 import \_3gpp-5gc-nrm-upffunction { prefix upf3gpp; }

 import \_3gpp-5gc-nrm-ddnmffunction { prefix ddnmff3gpp; }

 import \_3gpp-common-top { prefix top3gpp; }

 import ietf-inet-types { prefix inet; }

 organization "3GPP SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "Defines the YANG mapping of the 5GC related endpoint

 Information Object Classes (IOCs) that are part of the 5G Core

 Network Resource Model.";

 reference "3GPP TS 28.541";

 revision 2021-11-01 { reference CR-0614 ; }

 revision 2019-11-18 {

 reference "Ericsson refactoring.";

 }

 revision 2018-07-31 {

 reference "Initial revision";

 }

 grouping EP\_N2Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N3Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N4Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N5Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N6Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N64Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N65Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N66Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N7Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N8Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N9Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N10Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N11Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N12Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N13Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N14Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N15Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N16Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N17Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N20Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N21Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N22Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N26Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N27Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N31Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N32Grp {

 uses eprp3gpp:EP\_Common;

 leaf remoteSeppAddress {

 description "The host address of the SEPP.";

 type inet:host;

 }

 leaf remoteSeppId {

 type uint16;

 }

 leaf n32cParas {

 type string;

 }

 leaf n32fPolicy {

 type string;

 }

 leaf withIPX {

 type boolean;

 }

 }

 grouping EP\_N33Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N70Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_N71Grp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_S5CGrp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_S5UGrp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_RxGrp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_MAP\_SMSCGrp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_NLSGrp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_NLGGrp {

 uses eprp3gpp:EP\_Common;

 }

 grouping EP\_SBI\_IPXGrp {

 uses eprp3gpp:EP\_Common;

 leaf-list sBIService {

 min-elements 1;

 config false;

 type string;

 }

 }

 augment "/me3gpp:ManagedElement/af3gpp:AFFunction" {

 list EP\_N6 {

 description "Represents the EP\_N6 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N6Grp;

 }

 }

 list EP\_Rx {

 description "Represents the EP\_Rx IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_RxGrp;

 }

 }

 }

 augment "/me3gpp:ManagedElement/amf3gpp:AMFFunction" {

 list EP\_N2 {

 description "Represents the EP\_N2 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N2Grp;

 }

 }

 list EP\_N8 {

 description "Represents the EP\_N8 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N8Grp;

 }

 }

 list EP\_N11 {

 description "Represents the EP\_N11 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N11Grp;

 }

 }

 list EP\_N12 {

 description "Represents the EP\_N12 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N12Grp;

 }

 }

 list EP\_N14 {

 description "Represents the EP\_N14 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N14Grp;

 }

 }

 list EP\_N15 {

 description "Represents the EP\_N15 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N15Grp;

 }

 }

 list EP\_N17 {

 description "Represents the EP\_N17 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N17Grp;

 }

 }

 list EP\_N20 {

 description "Represents the EP\_N20 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N20Grp;

 }

 }

 list EP\_N22 {

 description "Represents the EP\_N22 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N22Grp;

 }

 }

 list EP\_N26 {

 description "Represents the EP\_N26 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N26Grp;

 }

 }

 list EP\_N33 {

 description "Represents the EP\_N33 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N33Grp;

 }

 }

 list EP\_NLS {

 description "Represents the EP\_NLS IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_NLSGrp;

 }

 }

 list EP\_NLG {

 description "Represents the EP\_NLG IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_NLGGrp;

 }

 }

 }

 augment "/me3gpp:ManagedElement/ausf3gpp:AUSFFunction" {

 list EP\_N12 {

 description "Represents the EP\_N12 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N12Grp;

 }

 }

 list EP\_N13 {

 description "Represents the EP\_N13 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N13Grp;

 }

 }

 }

 augment "/me3gpp:ManagedElement/dn3gpp:DNFunction" {

 list EP\_N6 {

 description "Represents the EP\_N6 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N6Grp;

 }

 }

 }

 augment "/me3gpp:ManagedElement/lmf3gpp:LMFFunction" {

 list EP\_NLS {

 description "Represents the EP\_NLS IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_NLSGrp;

 }

 }

 }

 augment "/me3gpp:ManagedElement/n3iwf3gpp:N3IWFFunction" {

 list EP\_N2 {

 description "Represents the EP\_N2 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N2Grp;

 }

 }

 list EP\_N3 {

 description "Represents the EP\_N3 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N3Grp;

 }

 }

 }

 augment "/me3gpp:ManagedElement/ngeir3gpp:NGEIRFunction" {

 list EP\_N17 {

 description "Represents the EP\_N17 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N17Grp;

 }

 }

 }

 augment "/me3gpp:ManagedElement/nrf3gpp:NRFFunction" {

 list EP\_N27 {

 description "Represents the EP\_N27 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N26Grp;

 }

 }

 }

 augment "/me3gpp:ManagedElement/nssf3gpp:NSSFFunction" {

 list EP\_N22 {

 description "Represents the EP\_N22 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N22Grp;

 }

 }

 list EP\_N31 {

 description "Represents the EP\_N31 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N31Grp;

 }

 }

 }

 augment "/me3gpp:ManagedElement/pcf3gpp:PCFFunction" {

 list EP\_N5 {

 description "Represents the EP\_N5 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N5Grp;

 }

 }

 list EP\_N7 {

 description "Represents the EP\_N7 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N7Grp;

 }

 }

 list EP\_N15 {

 description "Represents the EP\_N15 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N15Grp;

 }

 }

 list EP\_N16 {

 description "Represents the EP\_N16 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N16Grp;

 }

 }

 list EP\_Rx {

 description "Represents the EP\_Rx IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_RxGrp;

 }

 }

 }

 augment "/me3gpp:ManagedElement/sepp3gpp:SEPPFunction" {

 list EP\_N32 {

 description "Represents the EP\_N32 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N32Grp;

 }

 }

 }

 augment "/me3gpp:ManagedElement/smsf3gpp:SMSFFunction" {

 list EP\_N20 {

 description "Represents the EP\_20 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N20Grp;

 }

 }

 list EP\_N21 {

 description "Represents the EP\_N21 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N21Grp;

 }

 }

 list EP\_MAP\_SMSC {

 description "Represents the EP\_MAP\_SMSC IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_MAP\_SMSCGrp;

 }

 }

 }

 augment "/me3gpp:ManagedElement/smf3gpp:SMFFunction" {

 list EP\_N4 {

 description "Represents the EP\_N4 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N4Grp;

 }

 }

 list EP\_N7 {

 description "Represents the EP\_N7 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N7Grp;

 }

 }

 list EP\_N10 {

 description "Represents the EP\_N10 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N10Grp;

 }

 }

 list EP\_N11 {

 description "Represents the EP\_N11 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N11Grp;

 }

 }

 list EP\_N16 {

 description "Represents the EP\_N16 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N16Grp;

 }

 }

 list EP\_S5C {

 description "Represents the EP\_S5C IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_S5CGrp;

 }

 }

 }

 augment "/me3gpp:ManagedElement/udm3gpp:UDMFunction" {

 list EP\_N8 {

 description "Represents the EP\_N8 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N8Grp;

 }

 }

 list EP\_N10 {

 description "Represents the EP\_N10 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N10Grp;

 }

 }

 list EP\_N13 {

 description "Represents the EP\_N13 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N13Grp;

 }

 }

 list EP\_N64 {

 description "This IOC represents the N64 interface between the

 UDM and 5G DDNMF, which is defined in 3GPP TS 23.304 [xx].";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N64Grp;

 }

 }

 }

 augment "/me3gpp:ManagedElement/upf3gpp:UPFFunction" {

 list EP\_N4 {

 description "Represents the EP\_N4 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N4Grp;

 }

 }

 list EP\_N3 {

 description "Represents the EP\_N3 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N3Grp;

 }

 }

 list EP\_N9 {

 description "Represents the EP\_N9 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N9Grp;

 }

 }

 list EP\_S5U {

 description "Represents the EP\_S5U IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_S5UGrp;

 }

 }

 list EP\_N6 {

 description "Represents the EP\_N6 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N6Grp;

 }

 }

 }

/\* TODO: HSSFunction not modeled. Where should these go?

 augment "/me3gpp:ManagedElement/hss3gpp:HSSFunction???" {

 list EP\_N70 {

 description "Represents the EP\_N70 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N70Grp;

 }

 }

 list EP\_N71 {

 description "Represents the EP\_N71 IOC.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N71Grp;

 }

 }

 }

 \*/

 augment "/me3gpp:ManagedElement/ddnmff3gpp:DDNMFFunction" {

 list EP\_N65 {

 description "This IOC represents the N65 interface between the

 5G DDNMF in the HPLMN and the 5G DDNMF in a Local PLMN

 (5G ProSe Direct Discovery), which is defined in

 TS 23.304 [73].";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N65Grp;

 }

 }

 list EP\_N66 {

 description "This IOC represents the N66 interface between the

 5G DDNMF in the HPLMN and the 5G DDNMF in the VPLMN, which

 is defined in TS 23.304 [73].";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses EP\_N66Grp;

 }

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **5th Change** |

## H.5.10 module \_3gpp-5gc-nrm-lmffunction.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-lmffunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-lmffunction;

 prefix lmf3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import \_3gpp-common-top { prefix top3gpp; }

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 organization "3gpp SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "This IOC represents the LMF function defined in 3GPP TS 23.501.";

 reference "3GPP TS 28.541";

 revision 2021-11-01 { reference Refactoring ; }

 revision 2019-10-25 { reference "S5-194457 S5193518"; }

 revision 2019-05-15 {

 description "initial revision";

 reference "Based on

 3GPP TS 28.541 V15.X.XX";

 }

 grouping LMFFunctionGrp {

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNIdList {

 description "List of at most six entries of PLMN Identifiers,

 but at least one (the primary PLMN Id).

 The PLMN Identifier is composed of a Mobile Country Code (MCC)

 and a Mobile Network Code (MNC).";

 min-elements 1;

 max-elements 6;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 list managedNFProfile {

 description "Profile definition of a Managed NF";

 key idx;

 min-elements 1;

 uses nfp3gpp:ManagedNFProfile;

 }

 list commModelList {

 description "Specifies a list of commModel. It can be used by NF and

 NF services to interact with each other in 5G Core network ";

 reference "3GPP TS 23.501";

 min-elements 1;

 key "groupId";

 uses types5g3gpp:CommModel;

 }

 }

 augment "/me3gpp:ManagedElement" {

 list LMFFunction {

 description "5G Core LMF Function";

 reference "3GPP TS 28.541";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses LMFFunctionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses;

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **6th Change** |

## H.5.12 module \_3gpp-5gc-nrm-nfprofile.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-nfprofile {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-nfprofile;

 prefix nfp3gpp;

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import ietf-inet-types { prefix inet; }

 import ietf-yang-types { prefix yang; }

 import \_3gpp-5gc-nrm-nfservice { prefix nfs3gpp; }

 organization "3gpp SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "NF profile class.";

 reference "3GPP TS 29.510";

 revision 2021-11-01 { reference Refactoring ; }

 revision 2019-06-17 {

 reference "initial revision";

 }

 grouping IpEndPoint {

 leaf hostAddr {

 type inet:host;

 }

 leaf transport {

 description "Transport protocol";

 type enumeration {

 enum TCP;

 }

 }

 leaf port {

 description "This parameter indicates the port number.";

 type inet:port-number;

 }

 }

 grouping SCPDomainInfo {

 leaf scpFQDN {

 description "FQDN of the SCP.";

 type inet:domain-name;

 }

 list scpEndPoints {

 description "IP address(es) and port information of the SCP.

 If port information is present in this attribute, it

 applies to any scheme (i.e. HTTP and HTTPS).";

 min-elements 1;

 key idx;

 leaf idx { type uint32; }

 uses IpEndPoint;

 }

 leaf-list scpPorts {

 description "SCP port number(s) for HTTP and/or HTTPS.

 This attribute shall be present if the SCP uses non-default

 HTTP and/or HTTPS ports and if the SCP does not provision

 port information within ScpDomainInfo for each SCP domain

 it belongs to.";

 min-elements 1;

 type inet:port-number;

 }

 leaf scpPrefix {

 description "Optional deployment specific string used to

 construct the apiRoot of the next hop SCP.";

 reference "Clause 6.10 of 3GPP TS 29.500";

 type string;

 }

 }

 grouping SCPInfoGrp {

 list scpDomainInfoList {

 description "SCP domain specific information of the SCP that

 differs from the common information in NFProfile data type.";

 min-elements 1;

 key idx;

 leaf idx { type uint32; }

 uses SCPDomainInfo;

 }

 leaf scpPrefix {

 description "Optional deployment specific string used to

 construct the apiRoot of the next hop SCP, as described

 in clause 6.10 of 3GPP TS 29.500.";

 type string;

 }

 leaf-list scpPorts {

 description "SCP port number(s) for HTTP and/or HTTPS.

 This attribute shall be present if the SCP uses non-default

 HTTP and/or HTTPS ports and if the SCP does not provision

 port information within ScpDomainInfo for each SCP domain it

 belongs to.";

 type inet:port-number;

 }

 leaf-list addressDomains {

 description "Pattern (regular expression according to the

 ECMA-262 dialect) representing the address domain names

 reachable through the SCP.

 Absence of this IE indicates the SCP can reach any address

 domain names in the SCP domain(s) it belongs to.";

 min-elements 1;

 type string;

 }

 leaf-list ipv4Addresses {

 description "List of IPv4 addresses reachable through the SCP.

 If IPv4 addresses are reachable via the SCP, the absence of

 both this parameter and the ipv4AddrRanges one, indicates

 that the SCP can reach any IPv4 address in the SCP domain(s)

 it belongs to.";

 min-elements 1;

 type inet:ipv4-address;

 }

 leaf-list ipv6Prefixes {

 description "List of IPv6 prefixes reachable through the SCP.

 If IPv6 addresses are reachable via the SCP, the absence of

 both this parameter and the ipv6PrefixRanges one indicates

 the SCP can reach any IPv6 prefixes in the SCP domain(s)

 it belongs to.";

 min-elements 1;

 type inet:ipv6-prefix;

 }

 list ipv4AddrRanges {

 description "List of IPv4 address ranges reachable through the SCP.

 If IPv4 addresses are reachable via the SCP, the absence of both

 this parameter and the ipv4AddrRanges one, indicates that the SCP

 can reach any IPv4 address in the SCP domain(s) it belongs to.";

 key "start end";

 min-elements 1;

 uses types3gpp:Ipv4AddressRange;

 }

 list ipv6PrefixRanges {

 description "List of IPv6 prefixes ranges reachable through the SCP.

 If IPv6 addresses are reachable via the SCP, absence of both this

 parameter and the ipv6Prefixes one indicates the SCP can reach any

 IPv6 prefixes in the SCP domain(s) it belongs to.";

 key "start end";

 uses types3gpp:Ipv6PrefixRange;

 }

 leaf-list servedNfSetIdList {

 description "List of NF set ID of NFs served by the SCP.

 Absence of this parameter indicates the SCP can reach

 any NF set in the SCP domain(s) it belongs to.";

 type string;

 }

 list remotePlmnList {

 description "List of remote PLMNs reachable through the SCP.

 Absence of this parameter indicates that no remote PLMN is

 reachable through the SCP.";

 key "mcc mnc";

 min-elements 1;

 uses types3gpp:PLMNId;

 }

 leaf ipReachability {

 description "Indicates the type(s) of IP addresses reachable

 via the SCP in the SCP domain(s) it belongs to.";

 type enumeration {

 enum IPV4;

 enum IPV6;

 enum IPV4V6;

 }

 }

 }

 grouping ManagedNFProfile {

 description "Defines profile for managed NF";

 reference "3GPP TS 23.501";

 leaf idx { type uint32 ; }

 leaf nfInstanceID {

 config false;

 mandatory true;

 type yang:uuid ;

 description "This parameter defines profile for managed NF.

 The format of the NF Instance ID shall be a

 Universally Unique Identifier (UUID) version 4,

 as described in IETF RFC 4122 " ;

 }

 leaf-list nfType {

 config false;

 min-elements 1;

 type types3gpp:NfType;

 description "Type of the Network Function" ;

 }

 leaf hostAddr {

 mandatory true;

 type inet:host ;

 description "Host address of a NF";

 }

 leaf authzInfo {

 type string ;

 description "This parameter defines NF Specific Service authorization

 information. It shall include the NF type (s) and NF realms/origins

 allowed to consume NF Service(s) of NF Service Producer.";

 reference "See TS 23.501" ;

 }

 leaf locality {

 type string ;

 description "Information about the location of the NF instance

 (e.g. geographic location, data center) defined by operator";

 reference "TS 29.510" ;

 }

 leaf capacity {

 mandatory true;

 type uint16 ;

 description "This parameter defines static capacity information

 in the range of 0-65535, expressed as a weight relative to other

 NF instances of the same type; if capacity is also present in the

 nfServiceList parameters, those will have precedence over this value.";

 reference "TS 29.510" ;

 }

 leaf-list scpDomains {

 description "This parameter shall carry the list of SCP domains

 the SCP belongs to, or the SCP domain the NF (other than SCP)

 or the SEPP belongs to.";

 min-elements 1;

 type string;

 }

 list scpInfo {

 description "Service Communication Proxy (SCP) instance.";

 reference "3GPP TS 29.510";

 min-elements 1;

 key idx;

 leaf idx { type uint32; }

 uses SCPInfoGrp;

 }

 leaf vendorId {

 description "Vendor ID of the NF instance, according to the

 IANA-assigned 'SMI Network Management Private Enterprise Codes'.

 6 decimal digits; if the SMI code has less than 6 digits,

 it shall be padded with leading digits '0' to complete a

 6-digit string value.";

 type string {

 pattern "[0-9]{6}";

 }

 }

 list nFInfo {

 description "This parameter includes NF specific data in

 Managed NF profile";

 min-elements 1;

 max-elements 1;

 key idx;

 leaf idx { type uint32; }

 // 28.541 lists 5 possible cases with different children here:

 // case UdmInfo: nFSrvGroupId

 // case AusfInfo: nFSrvGroupId

 // case AmfInfo: not defined in 28.541, igroring for now

 // case UpfInfo: smfServingAreas

 // case UdrInfo: nFSrvGroupId, supportedDataSetIds

 must // UdmInfo, AusfInfo cases

 "(nFSrvGroupId and " +

 " not(smfServingAreas) and not(supportedDataSetIds)) or " +

 // UpfInfo case

 "(smfServingAreas and " +

 " not(nFSrvGroupId) and not(supportedDataSetIds)) or " +

 // UdrInfo case

 "(nFSrvGroupId and supportedDataSetIds and " +

 " not(smfServingAreas))";

 leaf nFSrvGroupId {

 description "This parameter defines identity of the group that

 is served by the NF instance";

 reference "TS 29.510";

 type string;

 }

 leaf-list smfServingAreas {

 description "This parameter defines the SMF service area(s)

 the UPF can serve";

 reference "TS 29.510";

 type string;

 }

 leaf-list supportedDataSetIds {

 description "Defines list of supported data sets in the

 UDR instance";

 reference "TS 29.510";

 type enumeration {

 enum SUBSCRIPTION;

 enum POLICY;

 enum EXPOSURE;

 enum APPLICATION;

 }

 }

 }

 }

 grouping NFProfileGrp {

 leaf nfInstanceID {

 description "String uniquely identifying a NF instance.";

 mandatory true;

 type string;

 }

 leaf nfType {

 description "Type of Network Function.";

 mandatory true;

 type types3gpp:NfType;

 }

 leaf nfStatus {

 description "Status of the NF Instance.";

 mandatory true;

 type NFStatus;

 }

 leaf heartBeatTimer {

 description "Time in seconds expected between 2 consecutive

 heart-beat messages from an NF Instance to the NRF. It may be

 included in the registration request.

 When present in the request it shall contain the heartbeat

 time proposed by the NF service consumer.";

 //conditional support

 type uint16;

 }

 list plmnList {

 description "PLMN(s) of the Network Function.

 This IE shall be present if this information is available for

 the NF. If not provided, PLMN ID(s) of the PLMN of the NRF

 are assumed for the NF.";

 //conditional support

 min-elements 1;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 list sNssais { //is the key unique

 description "S-NSSAIs of the Network Function. If not provided,

 the NF can serve any S-NSSAI. When present this IE represents

 the list of S-NSSAIs supported in all the PLMNs listed in the

 plmnList IE.";

 min-elements 1;

 //optional support

 key "sst sd";

 uses Snssai;

 }

 list perPlmnSnssaiList {

 description "This IE may be included when the list of S-NSSAIs

 supported by the NF for each PLMN it is supporting is different.

 When present, this IE shall include the S-NSSAIs supported by

 the Network Function for each PLMN supported by the Network

 Function. When present, this IE shall override sNssais IE.";

 min-elements 1;

 //optional support

 key idx; //no obvious leaf to use as a key

 leaf idx { type uint32; }

 uses PlmnSnssai;

 }

 leaf-list nsiList {

 description "NSI identities of the Network Function.

 If not provided, the NF can serve any NSI.";

 //optional support

 min-elements 1;

 type string;

 }

 leaf fqdn {

 description "FQDN of the Network Function. For AMF, the FQDN

 registered with the NRF shall be that of the AMF Name.";

 //conditional support

 type inet:domain-name;

 }

 leaf interPlmnFqdn {

 description "If the NF needs to be discoverable by other NFs

 in a different PLMN, then an FQDN that is used for

 inter-PLMN routing is specified.";

 //conditional support

 type inet:domain-name;

 }

 leaf-list ipv4Addresses {

 description "IPv4 address(es) of the Network Function.";

 min-elements 1;

 //conditional support

 type inet:ipv4-address;

 }

 leaf-list ipv6Addresses {

 description "IPv6 address(es) of the Network Function.";

 min-elements 1;

 //conditional support

 type inet:ipv6-address;

 }

 list allowedPlmns {

 description "PLMNs allowed to access the NF instance.

 If not provided, any PLMN is allowed to access the NF.";

 min-elements 1;

 //optional support

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 leaf-list allowedNfTypes {

 description "Type of the NFs allowed to access the NF instance.

 If not provided, any NF type is allowed to access the NF.";

 min-elements 1;

 //optional support

 type types3gpp:NfType;

 }

 leaf-list allowedNfDomains {

 description "Pattern representing the NF domain names allowed

 to access the NF instance. If not provided, any NF domain

 is allowed to access the NF.";

 min-elements 1;

 //optional support

 type string;

 }

 list allowedNssais { //is the key unique

 description "S-NSSAI of the allowed slices to access the NF

 instance. If not provided, any slice is allowed to access the NF.";

 min-elements 1;

 //optional support

 key "sst sd";

 uses Snssai;

 }

 leaf priority {

 description "Priority (relative to other NFs of the same type)

 in the range of 0-65535, to be used for NF selection;

 lower values indicate a higher priority. If priority is also

 present in the nfServiceList parameters, those will have

 precedence over this value. The NRF may overwrite the received

 priority value when exposing an NFProfile with the

 Nnrf\_NFDiscovery service.";

 //optional support

 type uint16;

 }

 leaf capacity {

 description "Static capacity information in the range of 0-65535,

 expressed as a weight relative to other NF instances of the

 same type; if capacity is also present in the nfServiceList

 parameters, those will have precedence over this value.";

 //optional support

 type uint16;

 }

 leaf load {

 description "Dynamic load information, ranged from 0 to 100,

 indicates the current load percentage of the NF.";

 //optional support

 type types3gpp:Load;

 }

 leaf locality {

 description "Operator defined information about the location

 of the NF instance (e.g. geographic location, data center).";

 //optional support

 type string;

 }

 grouping udrInfo {

 //optional support

 leaf groupId {

 description "Identity of the UDR group that is served by the

 UDR instance. If not provided, the UDR instance does not

 pertain to any UDR group.";

 //optional support

 type string;

 }

 list supiRanges {

 description "List of ranges of SUPI's whose profile data is

 available in the UDR instance.";

 key "start end pattern";

 min-elements 1;

 //optional support

 uses SupiRange;

 }

 list gpsiRanges {

 description "List of ranges of GPSIs whose profile data is

 available in the UDR instance.";

 key "start end pattern";

 min-elements 1;

 //optional support

 uses IdentityRange;

 }

 list externalGroupIdentifiersRanges {

 description "List of ranges of external groups whose profile

 data is available in the UDR instance.";

 key "start end pattern";

 min-elements 1;

 //optional support

 uses IdentityRange;

 }

 leaf-list supportedDataSets {

 description "List of supported data sets in the UDR instance.

 If not provided, the UDR supports all data sets.";

 min-elements 1;

 //optional support

 type DataSetId;

 }

 }

 grouping udmInfo {

 //optional support

 leaf groupId {

 description "Identity of the UDM group that is served by the

 UDM instance. If not provided, the UDM instance does not

 pertain to any UDM group.";

 //optional support

 type string;

 }

 list supiRanges {

 description "List of ranges of SUPI's whose profile data is

 available in the UDM instance.";

 key "start end pattern";

 min-elements 1;

 //optional support

 uses SupiRange;

 }

 list gpsiRanges {

 description "List of ranges of GPSIs whose profile data is

 available in the UDM instance.";

 key "start end pattern";

 min-elements 1;

 //optional support

 uses IdentityRange;

 }

 list externalGroupIdentifiersRanges {

 description "List of ranges of external groups whose profile

 data is available in the UDM instance.";

 key "start end pattern";

 min-elements 1;

 //optional support

 uses IdentityRange;

 }

 leaf-list routingIndicators {

 description "List of Routing Indicator information that allows

 to route network signalling with SUCI to the UDM instance.

 If not provided, the UDM can serve any Routing Indicator.";

 //optional support

 min-elements 1;

 type string {

 pattern '[0-9]{1,4}';

 }

 }

 }

 grouping ausfInfo {

 //optional support

 leaf groupId {

 description "Identity of the AUSF group. If not provided,

 the AUSF instance does not pertain to any AUSF group.";

 //optional support

 type string;

 }

 list supiRanges {

 description "List of ranges of SUPIs that can be served by the

 AUSF instance. If not provided, the AUSF can serve any SUPI.";

 key "start end pattern";

 min-elements 1;

 //optional support

 uses SupiRange;

 }

 leaf-list routingIndicators {

 description "List of Routing Indicator information that allows

 to route network signalling with SUCI to the AUSF instance.

 If not provided, the AUSF can serve any Routing Indicator.";

 //optional support

 min-elements 1;

 type string {

 pattern '[0-9]{1,4}';

 }

 }

 }

 grouping amfInfo {

 //optional support

 leaf amfRegionId {

 description "AMF region identifier";

 type string;

 }

 leaf amfSetId {

 description "AMF set identifier";

 type string;

 }

 list guamiList {

 description "List of supported GUAMIs.";

 key idx; //no obvious leaf to use as a key

 leaf idx { type uint32; }

 min-elements 1;

 uses Guami;

 }

 list taiList {

 description "The list of TAIs the AMF can serve. It may

 contain the non-3GPP access TAI. The absence of this

 attribute and the taiRangeList attribute indicate that

 the AMF can be selected for any TAI in the serving network.";

 key idx; //no obvious leaf to use as a key

 leaf idx { type uint32; }

 //optional support

 min-elements 1;

 uses Tai;

 }

 list taiRangeList {

 description "The range of TAIs the AMF can serve. The absence

 of this attribute and the taiList attribute indicate that the

 AMF can be selected for any TAI in the serving network.";

 //optional support

 min-elements 1;

 key idx; //no obvious leaf to use as a key

 leaf idx { type uint32; }

 uses TaiRange;

 }

 list backupInfoAmfFailure {

 description "List of GUAMIs for which the AMF acts as a

 backup for AMF failure.";

 key idx; //no obvious leaf to use as a key

 leaf idx { type uint32; }

 //optional support

 min-elements 1;

 uses Guami;

 }

 list backupInfoAmfRemoval {

 description "List of GUAMIs for which the AMF acts as a

 backup for planned AMF removal.";

 key idx; //no obvious leaf to use as a key

 leaf idx { type uint32; }

 //optional support

 min-elements 1;

 uses Guami;

 }

 list n2InterfaceAmfInfo {

 description "N2 interface information of the AMF. This

 information needs not be sent in NF Discovery responses.

 It may be used by the NRF to update the DNS for AMF

 discovery by the 5G Access Network.";

 //optional support

 max-elements 1;

 key idx; //no obvious leaf to use as a key

 leaf idx { type uint32; }

 uses N2InterfaceAmfInfo;

 }

 }

 grouping smfInfo {

 //optional support

 list sNssaiSmfInfoList {

 description "List of parameters supported by the SMF per S-NSSAI.";

 min-elements 1;

 key idx; //no obvious leaf to use as a key

 leaf idx { type uint32; }

 uses sNssaiSmfInfoItem;

 }

 list taiList {

 description "The list of TAIs the SMF can serve. It may contain

 the non-3GPP access TAI. The absence of this attribute and

 the taiRangeList attribute indicate that the SMF can be selected

 for any TAI in the serving network.";

 key idx; //no obvious leaf to use as a key

 leaf idx { type uint32; }

 //optional support

 min-elements 1;

 uses Tai;

 }

 list taiRangeList {

 description "The range of TAIs the SMF can serve. The absence

 of this attribute and the taiList attribute indicate that

 the SMF can be selected for any TAI in the serving network.";

 //optional support

 min-elements 1;

 key idx; //no obvious leaf to use as a key

 leaf idx { type uint32; }

 uses TaiRange;

 }

 leaf pgwFqdn {

 description "The FQDN of the PGW if the SMF is a combined SMF/PGW-C.";

 //optional support

 type inet:domain-name;

 }

 leaf-list accessType {

 description "If included, this IE shall contain the access type

 (3GPP\_ACCESS and/or NON\_3GPP\_ACCESS) supported by the SMF.

 If not included, it shall be assumed the both access types are

 supported.";

 //conditional support

 min-elements 1;

 max-elements 2;

 type AccessType;

 }

 }

 grouping upfInfo {

 //optional support

 list sNssaiUpfInfoList {

 description "List of parameters supported by the UPF per S-NSSAI.";

 min-elements 1;

 key idx; //no obvious leaf to use as a key

 leaf idx { type uint32; }

 uses SnssaiUpfInfoItem;

 }

 leaf-list smfServingArea {

 description "The SMF service area(s) the UPF can serve.

 If not provided, the UPF can serve any SMF service area.";

 //optional support

 min-elements 1;

 type string;

 }

 list interfaceUpfInfo {

 description "List of User Plane interfaces configured on the UPF.

 When this IE is provided in the NF Discovery response, the NF

 Service Consumer (e.g. SMF) may use this information for UPF

 selection.";

 key idx; //no obvious leaf to use as a key

 leaf idx { type uint32; }

 //optional support

 min-elements 1;

 uses InterfaceUpfInfoItem;

 }

 leaf iwkEpsInd {

 description "Indicates whether interworking with EPS is supported

 by the UPF.

 true: Supported

 false (default): Not Supported";

 //optional support

 type boolean;

 }

 leaf-list pduSessionTypes {

 description "List of PDU session type(s) supported by the UPF.

 The absence of this attribute indicates that the UPF can be

 selected for any PDU session type.";

 //optional support

 min-elements 1;

 type PduSessionType;

 }

 }

 grouping pcfInfo {

 //optional support

 leaf-list dnnList {

 description "DNNs supported by the PCF.

 If not provided, the PCF can serve any DNN.";

 //optional support

 min-elements 1;

 type string;

 }

 list supiRanges {

 description "List of ranges of SUPIs that can be served by the

 PCF instance. If not provided, the PCF can serve any SUPI.";

 key "start end pattern";

 min-elements 1;

 //optional support

 uses SupiRange;

 }

 leaf rxDiamHost {

 description "This IE shall be present if the PCF supports Rx

 interface. When present, this IE shall indicate the Diameter

 host of the Rx interface for the PCF.";

 //conditional support

 type string {

 pattern '([A-Za-z0-9]+(-[A-Za-z0-9]+).)+[a-z]{2,}';

 }

 }

 leaf rxDiamRealm {

 description "This IE shall be present if the PCF supports Rx

 interface. When present, this IE shall indicate the Diameter

 realm of the Rx interface for the PCF.";

 //conditional support

 type string {

 pattern '([A-Za-z0-9]+(-[A-Za-z0-9]+).)+[a-z]{2,}';

 }

 }

 }

 grouping bsfInfo {

 //optional support

 list ipv4AddressRanges {

 description "List of ranges of IPv4 addresses handled by BSF.

 If not provided, the BSF can serve any IPv4 address.";

 //optional support

 key "start end";

 uses types3gpp:Ipv4AddressRange;

 }

 leaf-list dnnList {

 description "List of DNNs handled by the BSF

 If not provided, the BSF can serve any DNN.";

 //optional support

 min-elements 1;

 type string;

 }

 leaf-list ipDomainList {

 description "List of IPv4 address domains, as described in

 subclause 6.2 of 3GPP TS 29.513, handled by the BSF.

 If not provided, the BSF can serve any IP domain.";

 //optional support

 min-elements 1;

 type string;

 }

 list ipv6PrefixRanges {

 description "List of ranges of IPv6 prefixes handled by the BSF.

 If not provided, the BSF can serve any IPv6 prefix.";

 //optional support

 key "start end";

 uses types3gpp:Ipv6PrefixRange;

 }

 }

 grouping chfInfo {

 //optional support

 list supiRangeList {

 description "List of ranges of SUPIs that can be served by the

 CHF instance. If not provided, the CHF can serve any SUPI.";

 key "start end pattern";

 min-elements 1;

 //optional support

 uses SupiRange;

 }

 list gpsiRangeList {

 description "List of ranges of GPSI that can be served by the

 CHF instance. If not provided, the CHF can serve any GPSI.";

 key "start end pattern";

 min-elements 1;

 //optional support

 uses IdentityRange;

 }

 list plmnRangeList {

 description "List of ranges of PLMNs (including the PLMN IDs of

 the CHF instance) that can be served by the CHF instance.

 If not provided, the CHF can serve any PLMN.";

 min-elements 1;

 //optional support

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 }

 grouping nrfInfoGrp {

 //optional support

 list servedUdrInfo {

 description "This attribute contains all the udrInfo attributes

 locally configured in the NRF or the NRF received during NF

 registration.";

 //optional support

 key nfInstanceID;

 leaf nfInstanceID {

 description "String uniquely identifying a NF instance.";

 type string;

 }

 min-elements 1;

 uses udrInfo;

 }

 list servedUdmInfo {

 description "This attribute contains all the udmInfo attributes

 locally configured in the NRF or the NRF received during NF

 registration.";

 //optional support

 key nfInstanceID;

 leaf nfInstanceID {

 description "String uniquely identifying a NF instance.";

 type string;

 }

 min-elements 1;

 uses udmInfo;

 }

 list servedAusfInfo {

 description "This attribute contains all the ausfInfo attributes

 locally configured in the NRF or the NRF received during NF

 registration.";

 //optional support

 key nfInstanceID;

 leaf nfInstanceID {

 description "String uniquely identifying a NF instance.";

 type string;

 }

 min-elements 1;

 uses ausfInfo;

 }

 list servedAmfInfo {

 description "This attribute contains all the amfInfo attributes

 locally configured in the NRF or the NRF received during NF

 registration.";

 //optional support

 key nfInstanceID;

 leaf nfInstanceID {

 description "String uniquely identifying a NF instance.";

 type string;

 }

 min-elements 1;

 uses amfInfo;

 }

 list servedSmfInfo {

 description "This attribute contains all the smfInfo attributes

 locally configured in the NRF or the NRF received during NF

 registration.";

 //optional support

 key nfInstanceID;

 leaf nfInstanceID {

 description "String uniquely identifying a NF instance.";

 type string;

 }

 min-elements 1;

 uses smfInfo;

 }

 list servedUpfInfo {

 description "This attribute contains all the upfInfo attributes

 locally configured in the NRF or the NRF received during NF

 registration.";

 //optional support

 key nfInstanceID;

 leaf nfInstanceID {

 description "String uniquely identifying a NF instance.";

 type string;

 }

 min-elements 1;

 uses upfInfo;

 }

 list servedPcfInfo {

 description "This attribute contains all the pcfInfo attributes

 locally configured in the NRF or the NRF received during NF

 registration.";

 //optional support

 key nfInstanceID;

 leaf nfInstanceID {

 description "String uniquely identifying a NF instance.";

 type string;

 }

 min-elements 1;

 uses pcfInfo;

 }

 list servedBsfInfo {

 description "This attribute contains all the bsfInfo attributes

 locally configured in the NRF or the NRF received during NF

 registration.";

 //optional support

 key nfInstanceID;

 leaf nfInstanceID {

 description "String uniquely identifying a NF instance.";

 type string;

 }

 min-elements 1;

 uses bsfInfo;

 }

 list servedChfInfo {

 description "This attribute contains all the bsfInfo attributes

 locally configured in the NRF or the NRF received during NF

 registration.";

 //optional support

 key nfInstanceID;

 leaf nfInstanceID {

 description "String uniquely identifying a NF instance.";

 type string;

 }

 min-elements 1;

 uses chfInfo;

 }

 }

 list nrfInfo {

 description "Information of an NRF NF Instance,

 used in hierarchical NRF deployments";

 key idx; //no obvious leaf to use as a key

 leaf idx { type uint32; }

 max-elements 1;

 uses nrfInfoGrp;

 }

 leaf customInfo {

 description "Specific data for custom Network Functions.";

 type string;

 }

 leaf recoveryTime {

 description "Timestamp when the NF was (re)started.";

 //optional support

 type yang:date-and-time;

 }

 leaf nfServicePersistence {

 description "If present, and set to true, it indicates that the

 different service instances of a same NF Service in this NF instance,

 supporting a same API version, are capable to persist their resource

 state in shared storage and therefore these resources are available

 after a new NF service instance supporting the same API version

 is selected by a NF Service Consumer (see 3GPP TS 23.527).

 Otherwise, it indicates that the NF Service Instances of a same

 NF Service are not capable to share resource state inside the

 NF Instance.";

 //optional support

 type boolean;

 }

 list nfServices {

 description "List of NF Service Instances. It shall include the

 services produced by the NF that can be discovered by other NFs.";

 key serviceInstanceID;

 //optional support

 min-elements 1;

 uses nfs3gpp:NFServiceGrp;

 }

 leaf nfProfileChangesSupportInd {

 description "NF Profile Changes Support Indicator. This IE may be

 present in the NFRegister or NFUpdate (NF Profile Complete

 Replacement) request and shall be absent in the response.

 true: the NF Service Consumer supports receiving NF Profile

 Changes in the response.

 false (default): the NF Service Consumer does not support

 receiving NF Profile Changes in the response.";

 //optional support

 type boolean;

 }

 leaf nfProfileChangesInd {

 description "NF Profile Changes Indicator. This IE shall be

 absent in the request to the NRF and may be included by the NRF

 in NFRegister or NFUpdate (NF Profile Complete Replacement)

 response.

 true: the NF Profile contains NF Profile changes.

 false (default): complete NF Profile.";

 //optional support

 type boolean;

 }

 list defaultNotificationSubscriptions {

 description "Notification endpoints for different notification types.";

 key notificationType;

 //optional support

 min-elements 1;

 uses types3gpp:DefaultNotificationSubscription;

 }

 }

 typedef NFStatus {

 type enumeration {

 enum REGISTERED;

 enum SUSPENDED;

 }

 }

 typedef DataSetId {

 type enumeration {

 enum SUBSCRIPTION;

 enum POLICY;

 enum EXPOSURE;

 enum APPLICATION;

 }

 }

 grouping SupiRange {

 leaf start {

 description "First value identifying the start of a SUPI range.

 To be used when the range of SUPI's can be represented as a

 numeric range (e.g., IMSI ranges).";

 type string {

 pattern '[0-9]+';

 }

 }

 leaf end {

 description "Last value identifying the end of a SUPI range.

 To be used when the range of SUPI's can be represented as a

 numeric range (e.g. IMSI ranges).";

 type string {

 pattern '[0-9]+';

 }

 }

 leaf pattern {

 description "Pattern representing the set of SUPI's belonging

 to this range. A SUPI value is considered part of the range

 if and only if the SUPI string fully matches the regular

 expression.";

 type string;

 }

 }

 grouping IdentityRange {

 leaf start {

 description "First value identifying the start of an identity

 range. To be used when the range of identities can be represented

 as a numeric range (e.g., MSISDN ranges).";

 type string {

 pattern '[0-9]+';

 }

 }

 leaf end {

 description "Last value identifying the end of an identity range.

 To be used when the range of identities can be represented as a

 numeric range (e.g. MSISDN ranges).";

 type string {

 pattern '[0-9]+';

 }

 }

 leaf pattern {

 description "Pattern representing the set of identities belonging

 to this range. An identity value is considered part of the range

 if and only if the identity string fully matches the regular

 expression.";

 type string;

 }

 }

 grouping TacRange {

 leaf start {

 description "First value identifying the start of a TAC range,

 to be used when the range of TAC's can be represented as a

 hexadecimal range (e.g., TAC ranges).";

 type string {

 pattern '([A-Fa-f0-9]{4}|[A-Fa-f0-9]{6})';

 }

 }

 leaf end {

 description "Last value identifying the end of a TAC range,

 to be used when the range of TAC's can be represented as a

 hexadecimal range (e.g. TAC ranges).";

 type string {

 pattern '([A-Fa-f0-9]{4}|[A-Fa-f0-9]{6})';

 }

 }

 leaf pattern {

 description "Pattern representing the set of TAC's belonging to

 this range.";

 type string;

 }

 }

 grouping SnssaiUpfInfoItem {

 list sNssai { //is the key unique

 description "Supported S-NSSAI.";

 min-elements 1;

 max-elements 1;

 key "sst sd";

 uses Snssai;

 }

 list dnnUpfInfoList {

 description "List of parameters supported by the UPF per DNN.";

 min-elements 1;

 key dnn;

 uses DnnUpfInfoItem;

 }

 }

 grouping DnnUpfInfoItem {

 leaf dnn {

 description "String representing a Data Network.";

 mandatory true;

 type string;

 }

 leaf-list dnaiList {

 description "List of Data network access identifiers supported

 by the UPF for this DNN. The absence of this attribute indicates

 that the UPF can be selected for this DNN for any DNAI.";

 min-elements 1;

 type string; // dnai is the type but its only a string with desc:

 // DNAI (Data network access identifier),

 // is this needed as its own typedef or string is ok

 }

 leaf-list pduSessionTypes {

 description "List of PDU session type(s) supported by the UPF

 for a specific DNN.";

 min-elements 1;

 type PduSessionType;

 }

 }

 grouping Snssai {

 leaf sst {

 description "Unsigned integer, within the range 0 to 255,

 representing the Slice/Service Type. It indicates the

 expected Network Slice behaviour in terms of features and services.";

 mandatory true;

 type uint32;

 }

 leaf sd {

 description "3-octet string, representing the Slice Differentiator,

 in hexadecimal representation.";

 //optional

 type string {

 pattern '[A-Fa-f0-9]{6}';

 }

 }

 reference "3GPP TS 29.571";

 }

 typedef PduSessionType {

 type enumeration {

 enum IPV4;

 enum IPV6;

 enum IPV4V6;

 enum UNSTRUCTURED;

 enum ETHERNET;

 }

 }

 grouping Guami {

 list plmnId {

 description "PLMN Identity.";

 min-elements 1;

 max-elements 1;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 list amfId {

 description "AMF Identity.";

 min-elements 1;

 max-elements 1;

 key "amfRegionId amfSetId amfPointer";

 uses types3gpp:AmfIdentifier;

 }

 }

 grouping Tai {

 list plmnId {

 description "PLMN Identity.";

 min-elements 1;

 max-elements 1;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 leaf tac { type types3gpp:Tac; }

 }

 grouping InterfaceUpfInfoItem {

 leaf interfaceType {

 description "User Plane interface type.";

 mandatory true;

 type UPInterfaceType;

 }

 ////At least one of the addressing parameters (ipv4address, ipv6adress

 // or endpointFqdn) shall be included in the InterfaceUpfInfoItem.

 choice address {

 case ipv4EndpointAddresses {

 leaf-list ipv4EndpointAddresses {

 description "Available endpoint IPv4 address(es) of the

 User Plane interface.";

 //conditional support

 min-elements 1;

 type inet:ipv4-address;

 }

 }

 case ipv6EndpointAddresses {

 leaf-list ipv6EndpointAddresses {

 description "Available endpoint IPv6 address(es) of the

 User Plane interface.";

 //conditional support

 min-elements 1;

 type inet:ipv6-address;

 }

 }

 case endpointFqdn {

 leaf endpointFqdn {

 description "FQDN of available endpoint of the

 User Plane interface.";

 //conditional support

 type inet:domain-name;

 }

 }

 }

 leaf networkInstance {

 description "Network Instance associated to the

 User Plane interface.";

 //optional support

 type string;

 }

 }

 typedef UPInterfaceType {

 type enumeration {

 enum N3;

 enum N6;

 enum N9;

 }

 }

 grouping TaiRange {

 list plmnId {

 description "PLMN ID related to the TacRange.";

 min-elements 1;

 max-elements 1;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 list tacRangeList { //is this key unique

 description "The range of the TACs.";

 min-elements 1;

 key "start end";

 uses TacRange;

 }

 }

 typedef AccessType {

 type enumeration {

 enum 3GPP\_ACCESS;

 enum NON\_3GPP\_ACCESS;

 }

 }

 grouping N2InterfaceAmfInfo {

 //At least one of the addressing parameters (ipv4address or

 // ipv6adress) shall be included.

 choice address {

 case ipv4EndpointAddress {

 leaf-list ipv4EndpointAddress {

 description "Available AMF endpoint IPv4 address(es) for N2.";

 //conditional support

 min-elements 1;

 type inet:ipv4-address;

 }

 }

 case ipv6EndpointAddress {

 leaf-list ipv6EndpointAddress {

 description "Available AMF endpoint IPv6 address(es) for N2.";

 //conditional support

 min-elements 1;

 type inet:ipv6-address;

 }

 }

 }

 leaf amfName {

 description "AMF name.";

 type string;

 }

 }

 grouping sNssaiSmfInfoItem {

 list sNssai { //is the key unique

 description "Supported S-NSSAI.";

 min-elements 1;

 max-elements 1;

 key "sst sd";

 uses Snssai;

 }

 list dnnSmfInfoList { //is the key unique

 description "List of parameters supported by the SMF per DNN.";

 min-elements 1;

 key dnn;

 uses DnnSmfInfoItem;

 }

 }

 grouping DnnSmfInfoItem {

 leaf dnn {

 description "Supported DNN.";

 mandatory true;

 type string;

 }

 }

 grouping PlmnSnssai {

 list plmnId {

 description "PLMN ID for which list of supported S-NSSAI(s)

 is provided.";

 min-elements 1;

 max-elements 1;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 list sNssaiList { //is the key unique

 description "The specific list of S-NSSAIs supported by the

 given PLMN.";

 min-elements 1;

 key "sst sd";

 uses Snssai;

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **7th Change** |

## H.5.14 module \_3gpp-5gc-nrm-ngeirfunction.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-ngeirfunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-ngeirfunction;

 prefix ngeir3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import \_3gpp-common-top { prefix top3gpp; }

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 organization "3gpp SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "This IOC represents the 5G-EIR function in 5GC. For more

 information about the 5G-EIR, see 3GPP TS 23.501.";

 reference "3GPP TS 28.541";

 revision 2021-11-01 { reference Refactoring ; }

 revision 2020-11-05 { reference CR-0412 ; }

 revision 2019-10-25 { reference "S5-194457 S5-195427 S5-193518"; }

 revision 2019-05-15 {reference "initial revision"; }

 grouping NGEIRFunctionGrp {

 description "Represents the NGEIRFunction IOC";

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNIdList {

 description "List of at most six entries of PLMN Identifiers, but at

 least one (the primary PLMN Id).

 The PLMN Identifier is composed of a Mobile Country Code (MCC) and

 a Mobile Network Code (MNC).";

 min-elements 1;

 max-elements 6;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 list sNSSAIList {

 description "List of S-NSSAIs the managed object is capable of supporting.

 (Single Network Slice Selection Assistance Information)

 An S-NSSAI has an SST (Slice/Service type) and an optional SD

 (Slice Differentiator) field.";

 //optional support

 reference "3GPP TS 23.003";

 key "sd sst";

 uses types5g3gpp:SNssai;

 }

 list managedNFProfile {

 description "Profile definition of a Managed NF";

 key idx;

 min-elements 1;

 max-elements 1;

 uses nfp3gpp:ManagedNFProfile;

 }

 list commModelList {

 min-elements 1;

 key "groupId";

 description "Specifies a list of commModel. It can be used by NF and

 NF services to interact with each other in 5G Core network ";

 reference "3GPP TS 23.501";

 uses types5g3gpp:CommModel;

 }

 }

 augment "/me3gpp:ManagedElement" {

 list NGEIRFunction {

 description "5G Core NGEIR Function";

 reference "3GPP TS 28.541";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses NGEIRFunctionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses;

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **8th Change** |

## H.5.16 module \_3gpp-5gc-nrm-nssffunction.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-nssffunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-nssffunction;

 prefix nssf3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import ietf-inet-types { prefix inet; }

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import \_3gpp-common-top { prefix top3gpp; }

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 organization "3gpp SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "This IOC represents the NSSF function in 5GC. For more

 information about the NSSF, see 3GPP TS 23.501.";

 reference "3GPP TS 28.541";

 revision 2021-11-01 { reference Refactoring ; }

 revision 2020-11-05 { reference CR-0412 ; }

 revision 2020-08-03 { reference "CR-0321"; }

 revision 2019-10-25 { reference "S5-194457 S5-195427 S5-193518"; }

 revision 2019-05-15 { reference "initial revision"; }

 grouping NSSFFunctionGrp {

 description "Represents the NSSFFunction IOC";

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNIdList {

 description "List of at most six entries of PLMN Identifiers, but at least

 one (the primary PLMN Id).

 The PLMN Identifier is composed of a Mobile Country Code (MCC) and a

 Mobile Network Code (MNC).";

 min-elements 1;

 max-elements 6;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 leaf sBIFQDN {

 description "The FQDN of the registered NF instance in the service-based

 interface.";

 type inet:domain-name;

 }

 list sNSSAIList {

 description "List of S-NSSAIs the managed object is capable of supporting.

 (Single Network Slice Selection Assistance Information)

 An S-NSSAI has an SST (Slice/Service type) and an optional SD

 (Slice Differentiator) field.";

 reference "3GPP TS 23.003";

 key "sd sst";

 uses types5g3gpp:SNssai;

 }

 leaf-list cNSIIdList {

 description "NSI ID. NSI ID is an identifier for identifying the Core

 Network part of a Network Slice instance when multiple Network Slice

 instances of the same Network Slice are deployed, and there is a need

 to differentiate between them in the 5GC, see clause 3.1 of TS 23.501

 and subclause 6.1.6.2.7 of 3GPP TS 29.531";

 type string;

 }

 list managedNFProfile {

 description "Profile definition of a Managed NF";

 key idx;

 min-elements 1;

 max-elements 1;

 uses nfp3gpp:ManagedNFProfile;

 }

 }

 augment "/me3gpp:ManagedElement" {

 list NSSFFunction {

 description "5G Core NSSF Function";

 reference "3GPP TS 28.541";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses NSSFFunctionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses;

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **9th Change** |

## H.5.17 module \_3gpp-5gc-nrm-nwdaffunction.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-nwdaffunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-nwdaffunction;

 prefix nwdaf3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import ietf-inet-types { prefix inet; }

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import \_3gpp-common-top { prefix top3gpp; }

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 organization "3gpp SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "This IOC represents the NWDAF function in 5GC.

 For more information about the NWDAF, see 3GPP TS 23.501.";

 reference "3GPP TS 28.541";

 revision 2021-11-01 { reference Refactoring ; }

 revision 2020-11-05 { reference CR-0412 ; }

 revision 2019-10-25 { reference "S5-194457 S5-195427 S5-193518"; }

 revision 2019-05-15 {reference "initial revision"; }

 grouping NWDAFFunctionGrp {

 description "Represents the NWDAFFunction IOC";

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNIdList {

 description "List of at most six entries of PLMN Identifiers, but at

 least one (the primary PLMN Id).

 The PLMN Identifier is composed of a Mobile Country Code (MCC) and a

 Mobile Network Code (MNC).";

 min-elements 1;

 max-elements 6;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 leaf sBIFQDN {

 description "The FQDN of the registered NF instance in the service-based

 interface.";

 type inet:domain-name;

 }

 list sNSSAIList {

 description "List of S-NSSAIs the managed object is capable of supporting.

 (Single Network Slice Selection Assistance Information)

 An S-NSSAI has an SST (Slice/Service type) and an optional SD

 (Slice Differentiator) field.";

 //optional support

 reference "3GPP TS 23.003";

 key "sd sst";

 uses types5g3gpp:SNssai;

 }

 list managedNFProfile {

 description "Profile definition of a Managed NF";

 key idx;

 min-elements 1;

 max-elements 1;

 uses nfp3gpp:ManagedNFProfile;

 }

 list commModelList {

 min-elements 1;

 key "groupId";

 description "Specifies a list of commModel. It can be used by NF and

 NF services to interact with each other in 5G Core network ";

 reference "3GPP TS 23.501";

 uses types5g3gpp:CommModel;

 }

 }

 augment "/me3gpp:ManagedElement" {

 list NWDAFFunction {

 description "5G Core NWDAF Function";

 reference "3GPP TS 28.541";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses NWDAFFunctionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses;

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **10th Change** |

## H.5.18 module \_3gpp-5gc-nrm-pcffunction.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-pcffunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-pcffunction;

 prefix pcf3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import ietf-inet-types { prefix inet; }

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import \_3gpp-common-top { prefix top3gpp; }

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 organization "3gpp SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "This IOC represents the PCF function in 5GC. For more

 information about the PCF, see 3GPP TS 23.501.";

 reference "3GPP TS 28.541";

 revision 2021-11-01 { reference Refactoring ; }

 revision 2020-11-05 { reference CR-0412 ; }

 revision 2020-08-06 { reference "CR-0333"; }

 revision 2020-08-06 { reference "CR-0331"; }

 revision 2019-10-25 { reference "S5-194457 S5-193518"; }

 revision 2019-05-22 { reference "initial revision"; }

 grouping PCFFuntionGrp {

 description "Represents the PCFFuntion IOC";

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNIdList {

 description "List of at most six entries of PLMN Identifiers, but at

 least one (the primary PLMN Id).

 The PLMN Identifier is composed of a Mobile Country Code (MCC) and a

 Mobile Network Code (MNC).";

 min-elements 1;

 max-elements 6;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 leaf sBIFQDN {

 description "The FQDN of the registered NF instance in the service-based

 interface.";

 type inet:domain-name;

 }

 list sNSSAIList {

 description "List of S-NSSAIs the managed object is capable of supporting.

 (Single Network Slice Selection Assistance Information)

 An S-NSSAI has an SST (Slice/Service type) and an optional SD

 (Slice Differentiator) field.";

 //optional support

 reference "3GPP TS 23.003";

 key "sd sst";

 uses types5g3gpp:SNssai;

 }

 list managedNFProfile {

 description "Profile definition of a Managed NF";

 key idx;

 min-elements 1;

 max-elements 1;

 uses nfp3gpp:ManagedNFProfile;

 }

 list commModelList {

 min-elements 1;

 key "groupId";

 description "Specifies a list of commModel. It can be used by NF and

 NF services to interact with each other in 5G Core network ";

 reference "3GPP TS 23.501";

 uses types5g3gpp:CommModel;

 }

 leaf dynamic5QISetRef {

 type types3gpp:DistinguishedName;

 description "DN of the Dynamic5QISet that the PCFFunction supports

 (is associated to).";

 }

 leaf configurable5QISetRef {

 type types3gpp:DistinguishedName;

 description "DN of the Configurable5QISet that the PCFFunction supports

 (is associated to).";

 }

 }

 augment "/me3gpp:ManagedElement" {

 list PCFFunction {

 description "5G Core PCF Function";

 reference "3GPP TS 28.541";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses PCFFuntionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses;

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **11th Change** |

## H.5.20 module \_3gpp-5gc-nrm-smffunction.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-smffunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-smffunction;

 prefix smf3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import ietf-inet-types { prefix inet; }

 import \_3gpp-common-top { prefix top3gpp; }

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 organization "3gpp SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "SMFFunction derived from basic ManagedFunction.";

 reference "3GPP TS 28.541";

 revision 2021-11-01 { reference Refactoring ; }

 revision 2020-11-05 { reference CR-0412 ; }

 revision 2020-08-06 { reference "CR-0333"; }

 revision 2020-06-03 { reference "CR-0286"; }

 revision 2019-10-25 { reference "S5-194457 S5-193518"; }

 revision 2019-05-31 {reference "Ericsson refactoring."; }

 revision 2018-08-07 { reference "Initial revision";}

 grouping SMFFunctionGrp {

 description "Represents the SMFFuntion IOC";

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNIdList {

 min-elements 1;

 description "A list of PLMN identifiers (Mobile Country Code and Mobile

 Network Code).";

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 leaf-list nRTACList {

 description "List of Tracking Area Codes (legacy TAC or extended TAC)

 where the represented management function is serving.";

 reference "TS 38.413 clause 9.3.3.10";

 min-elements 1;

 config false;

 type types3gpp:Tac;

 }

 leaf sBIFQDN {

 description "The FQDN of the registered NF instance in the service-based

 interface.";

 type inet:domain-name;

 }

 list sNSSAIList {

 description "List of S-NSSAIs the managed object is capable of supporting.

 (Single Network Slice Selection Assistance Information)

 An S-NSSAI has an SST (Slice/Service type) and an optional SD

 (Slice Differentiator) field.";

 reference "3GPP TS 23.003";

 key "sd sst";

 uses types5g3gpp:SNssai;

 }

 list managedNFProfile {

 description "Profile definition of a Managed NF";

 key idx;

 min-elements 1;

 max-elements 1;

 uses nfp3gpp:ManagedNFProfile;

 }

 list commModelList {

 min-elements 1;

 key "groupId";

 description "Specifies a list of commModel. It can be used by NF and

 NF services to interact with each other in 5G Core network ";

 reference "3GPP TS 23.501";

 uses types5g3gpp:CommModel;

 }

 leaf configurable5QISetRef {

 type types3gpp:DistinguishedName;

 description "DN of the Configurable5QISet that the SMFFunction supports

 (is associated to).";

 }

 leaf dynamic5QISetRef {

 type types3gpp:DistinguishedName;

 description "DN of the Dynamic5QISet that the SMFFunction supports

 (is associated to).";

 }

 }

 augment "/me3gpp:ManagedElement" {

 list SMFFunction {

 description "5G Core SMF Function";

 reference "3GPP TS 28.541";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses SMFFunctionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses;

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **12th Change** |

## H.5.21 module \_3gpp-5gc-nrm-smsffunction.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-smsffunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-smsffunction;

 prefix smsf3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import \_3gpp-common-top { prefix top3gpp; }

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 organization "3gpp SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "This IOC represents the SMSF function defined in 3GPP TS 23.501.";

 reference "3GPP TS 28.541";

 revision 2021-11-01 { reference Refactoring ; }

 revision 2019-10-25 { reference "S5-194457 S5-195427 S5-193518"; }

 revision 2019-05-15 {

 reference "initial revision";

 }

 grouping SMSFFunctionGrp {

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNIdList {

 description "List of at most six entries of PLMN Identifiers,

 but at least one (the primary PLMN Id).

 The PLMN Identifier is composed of a Mobile Country Code (MCC)

 and a Mobile Network Code (MNC).";

 min-elements 1;

 max-elements 6;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 list managedNFProfile {

 description "Profile definition of a Managed NF";

 key idx;

 min-elements 1;

 uses nfp3gpp:ManagedNFProfile;

 }

 list commModelList {

 description "Specifies a list of commModel. It can be used by NF and

 NF services to interact with each other in 5G Core network ";

 reference "3GPP TS 23.501";

 min-elements 1;

 key "groupId";

 uses types5g3gpp:CommModel;

 }

 }

 augment "/me3gpp:ManagedElement" {

 list SMSFFunction {

 description "5G Core SMSF Function";

 reference "3GPP TS 28.541";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses SMSFFunctionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses;

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **13th Change** |

## H.5.22 module \_3gpp-5gc-nrm-udmfunction.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-udmfunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-udmfunction;

 prefix udm3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import ietf-inet-types { prefix inet; }

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import \_3gpp-common-top { prefix top3gpp; }

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 organization "3gpp SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "This IOC represents the UDM function in 5GC. For more

 information about the UDM, see 3GPP TS 23.501.";

 reference "3GPP TS 28.541";

 revision 2021-11-01 { reference Refactoring ; }

 revision 2020-11-05 { reference CR-0412 ; }

 revision 2019-10-25 { reference "S5-194457 S5-195427 S5-193518"; }

 revision 2019-05-22 { reference "initial revision";}

 grouping UDMFuntionGrp {

 description "Represents the UDMFuntion IOC";

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNIdList {

 description "List of at most six entries of PLMN Identifiers, but at

 least one (the primary PLMN Id).

 The PLMN Identifier is composed of a Mobile Country Code (MCC) and a

 Mobile Network Code (MNC).";

 min-elements 1;

 max-elements 6;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 leaf sBIFQDN {

 description "The FQDN of the registered NF instance in the service-based

 interface.";

 type inet:domain-name;

 }

 list sNSSAIList {

 description "List of S-NSSAIs the managed object is capable of supporting.

 (Single Network Slice Selection Assistance Information)

 An S-NSSAI has an SST (Slice/Service type) and an optional SD

 (Slice Differentiator) field.";

 //optional support

 reference "3GPP TS 23.003";

 key "sd sst";

 uses types5g3gpp:SNssai;

 }

 list managedNFProfile {

 description "Profile definition of a Managed NF";

 key idx;

 min-elements 1;

 max-elements 1;

 uses nfp3gpp:ManagedNFProfile;

 }

 list commModelList {

 min-elements 1;

 key "groupId";

 description "Specifies a list of commModel. It can be used by NF and

 NF services to interact with each other in 5G Core network ";

 reference "3GPP TS 23.501";

 uses types5g3gpp:CommModel;

 }

 }

 augment "/me3gpp:ManagedElement" {

 list UDMFunction {

 description "5G Core UDM Function";

 reference "3GPP TS 28.541";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses UDMFuntionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses;

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **14th Change** |

## H.5.23 module \_3gpp-5gc-nrm-udrfunction.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-udrfunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-udrfunction;

 prefix udr3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import ietf-inet-types { prefix inet; }

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import \_3gpp-common-top { prefix top3gpp; }

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 organization "3gpp SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "This IOC represents the UDR function in 5GC. For more information

 about the UDR, see 3GPP TS 23.501.";

 reference "3GPP TS 28.541";

 revision 2021-11-01 { reference Refactoring ; }

 revision 2020-11-05 { reference CR-0412 ; }

 revision 2019-10-25 { reference "S5-194457 S5-195427 S5-193518"; }

 revision 2019-05-22 {reference "initial revision"; }

 grouping UDRFuntionGrp {

 description "Representse the UDRFuntion IOC";

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNIdList {

 description "List of at most six entries of PLMN Identifiers, but at

 least one (the primary PLMN Id).

 The PLMN Identifier is composed of a Mobile Country Code (MCC) and a

 Mobile Network Code (MNC).";

 min-elements 1;

 max-elements 6;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 leaf sBIFQDN {

 description "The FQDN of the registered NF instance in the service-based

 interface.";

 type inet:domain-name;

 }

 list sNSSAIList {

 description "List of S-NSSAIs the managed object is capable of supporting.

 (Single Network Slice Selection Assistance Information)

 An S-NSSAI has an SST (Slice/Service type) and an optional SD

 (Slice Differentiator) field.";

 //optional support

 reference "3GPP TS 23.003";

 key "sd sst";

 uses types5g3gpp:SNssai;

 }

 list managedNFProfile {

 description "Profile definition of a Managed NF";

 key idx;

 min-elements 1;

 max-elements 1;

 uses nfp3gpp:ManagedNFProfile;

 }

 }

 augment "/me3gpp:ManagedElement" {

 list UDRFunction {

 description "5G Core UDR Function";

 reference "3GPP TS 28.541";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses UDRFuntionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses;

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **15th Change** |

## H.5.24 module \_3gpp-5gc-nrm-udsffunction.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-udsffunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-udsffunction;

 prefix udsf3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import ietf-inet-types { prefix inet; }

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import \_3gpp-common-top { prefix top3gpp; }

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 organization "3gpp SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "This IOC represents the UDSF function which can be interacted

 with any other 5GC NF defined in 3GPP TS 23.501.";

 reference "3GPP TS 28.541";

 revision 2021-11-01 { reference Refactoring ; }

 revision 2020-11-05 { reference CR-0412 ; }

 revision 2019-10-25 { reference "S5-194457 S5-195427 S5-193518"; }

 revision 2019-05-22 { reference "initial revision"; }

 grouping UDSFFuntionGrp {

 description "Represents the UDSFFuntion IOC";

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNIdList {

 description "List of at most six entries of PLMN Identifiers, but at

 least one (the primary PLMN Id).

 The PLMN Identifier is composed of a Mobile Country Code (MCC) and a

 Mobile Network Code (MNC).";

 min-elements 1;

 max-elements 6;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 leaf sBIFQDN {

 description "The FQDN of the registered NF instance in the

 service-based interface.";

 type inet:domain-name;

 }

 list sNSSAIList {

 description "List of S-NSSAIs the managed object is capable of supporting.

 (Single Network Slice Selection Assistance Information)

 An S-NSSAI has an SST (Slice/Service type) and an optional SD

 (Slice Differentiator) field.";

 //optional support

 reference "3GPP TS 23.003";

 key "sd sst";

 uses types5g3gpp:SNssai;

 }

 list managedNFProfile {

 key idx;

 min-elements 1;

 max-elements 1;

 description "Managed Network Function profile";

 reference "3GPP TS 23.501";

 uses nfp3gpp:ManagedNFProfile;

 }

 }

 augment "/me3gpp:ManagedElement" {

 list UDSFFunction {

 description "5G Core UDSF Function";

 reference "3GPP TS 28.541";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses UDSFFuntionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses; }

 }

}

<CODE ENDS>

|  |
| --- |
| **16th Change** |

## H.5.25 module \_3gpp-5gc-nrm-upffunction.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-upffunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-upffunction;

 prefix upf3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import \_3gpp-common-top { prefix top3gpp; }

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 organization "3GPP SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "UPFFunction derived from basic ManagedFunction.";

 reference "3GPP TS 28.541 5G Network Resource Model (NRM)";

 revision 2021-11-01 { reference Refactoring ; }

 revision 2020-11-05 { reference CR-0412 ; }

 revision 2019-10-25 { reference "S5-194457 S5-193518"; }

 revision 2019-05-31 { reference "Ericsson refactoring."; }

 revision 2018-08-07 { reference "Initial revision"; }

 grouping UPFFunctionGrp {

 description "Represents the UPFFunction IOC";

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNIdList {

 description "A list of PLMN identifiers (Mobile Country Code and Mobile

 Network Code).";

 min-elements 1;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 leaf-list nRTACList {

 description "List of Tracking Area Codes (legacy TAC or extended TAC)

 where the represented management function is serving.";

 reference "TS 38.413 clause 9.3.3.10";

 min-elements 1;

 config false;

 type types3gpp:Tac;

 }

 list sNSSAIList {

 description "List of S-NSSAIs the managed object is capable of supporting.

 (Single Network Slice Selection Assistance Information)

 An S-NSSAI has an SST (Slice/Service type) and an optional SD

 (Slice Differentiator) field.";

 reference "3GPP TS 23.003";

 key "sd sst";

 uses types5g3gpp:SNssai;

 }

 list managedNFProfile {

 description "Profile definition of a Managed NF";

 key idx;

 min-elements 1;

 max-elements 1;

 reference "3GPP TS 23.003";

 uses nfp3gpp:ManagedNFProfile;

 }

 leaf-list supportedBMOList {

 type string;

 description "List of supported BMOs (Bridge Managed Objects) required

 for integration with TSN system.";

 }

 }

 augment /me3gpp:ManagedElement {

 list UPFFunction {

 description "5G Core UPF Function";

 reference "3GPP TS 28.541";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses UPFFunctionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses;

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **17th Change** |

## H.5.X module \_3gpp-5gc-nrm-ddnmffunction.yang

<CODE BEGINS>

module \_3gpp-5gc-nrm-ddnmffunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-ddnmffunction;

 prefix ddnmff3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

 import \_3gpp-common-yang-types { prefix types3gpp; }

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import ietf-inet-types { prefix inet; }

 import \_3gpp-common-top { prefix top3gpp; }

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 organization "3gpp SA5";

 contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

 description "DDNMFFunction derived from basic ManagedFunction.";

 reference "3GPP TS 28.541";

 revision 2021-11-01 { reference CR-0614 ; }

 grouping DDNMFFunctionGrp {

 description "

 This IOC represents the 5G DDNMF which is the logical function

 handling network related actions required for dynamic 5G ProSe

 Direct Discovery. For more information about the 5G DDNMF, see

 3GPP TS 23.304 [xx]. ";

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNIdList {

 min-elements 1;

 description "A list of PLMN identifiers (Mobile Country Code and Mobile

 Network Code).";

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 leaf sBIFQDN {

 description "The FQDN of the registered NF instance in the

 service-based interface.";

 type inet:domain-name;

 }

 list managedNFProfile {

 description "Profile definition of a Managed NF";

 reference "TS 23.501";

 key idx;

 min-elements 1;

 max-elements 1;

 uses nfp3gpp:ManagedNFProfile;

 }

 list commModelList {

 min-elements 1;

 key "groupId";

 description "Specifies a list of commModel. It can be used by NF and

 NF services to interact with each other in 5G Core network ";

 reference "3GPP TS 23.501";

 uses types5g3gpp:CommModel;

 }

 }

 augment "/me3gpp:ManagedElement" {

 list DDNMFFunction {

 description "5G Core DDNMF Function";

 reference "3GPP TS 28.541";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses DDNMFFunctionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses;

 }

 }

}

<CODE ENDS>

|  |
| --- |
| **End of Changes** |