**3GPP TSG-CT WG4 Meeting #110-eC4-223199**

**E-Meeting, 12th – 20th May 2022**

|  |
| --- |
| *CR-Form-v12.2* |
| **CHANGE REQUEST** |
|  |
|  | **29.571** | **CR** | **0362** | **rev** | **-** | **Current version:** | **17.5.0** |  |
|  |
| *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 |  | Core Network | **X** |

|  |
| --- |
|  |
| ***Title:***  | Provising of FQDN Matching Rule  |
|  |  |
| ***Source to WG:*** | Ericsson |
| ***Source to TSG:*** | C4 |
|  |  |
| ***Work item code:*** | eEDGE\_5GC |  | ***Date:*** | 2022-04-30 |
|  |  |  |  |  |
| ***Category:*** | **B** |  | ***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-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19)* |
|  |  |
| ***Reason for change:*** | CT3 and CT4 are including either arrays of FQDNs or FQDN regular expressions in their APIs in order to provide the receiver Network Function the list of FQDN(s) against which a matching is required. The use of arrays of FQDNs requires that the Network Function needs to provide each specific FQDN. This can be avoided with the use of regular expressions. However, this is considered an overkilled solution that, although regular expression is versatile and flexible, it may have a big impact in performance for UPF/EASDF and may cause faulty situations.Especially when a DNS Context or a PFCP session include multiple DNS matching template or PDRs using regular expression to match, it is possible that multiple templates or multiple PDRs get matched, **which leads completely different results than originally intended,** e.g. forward the DNS message to a wrong DNS server. These DNS Matching Templates in the DNS rule or Packet Detection Information in the PDRs are generated upon the request from DIFFERENT AFs, it would be heavy task or even impossible for SMF or PCF to validate and ensure there is no overlapping DNS template or PDRs when regular expression are used, the same for UPF/EASDF. There is a huge risk that an unexpected PDR or DNS template is matched and leads complete wrong result.This CR proposes to use a data type structure, FqdnMatchingRule, where a list of FQDNs to be matched is either described by a StringMatchingRule or Regular Expression, where the StringMatchingRule shall be used preferrably whenever possible to optimize the matching process and reduce processing load, e.g. in the UPF or EASDF.  |
|  |  |
| ***Summary of change:*** | Propose a new data type as alternative to a list of FQDN or using regular expression. |
|  |  |
| ***Consequences if not approved:*** | Misaligned specifications and suboptimal system design, high risk to lead error situation, e.g. forward a DNS request to a wrong DNS server. |
|  |  |
| ***Clauses affected:*** | 5.2.4.a, 5.2.4.x, 5.2.4.y, 5.2.3.x, A.2 |
|  |  |
|  | **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:*** | This CR is a backwards compatible feature in the OpenAPI file of the CommonData API. |
|  |  |
| ***This CR's revision history:*** | Rev1: Introducing FqdnMatchingRule which is either described by a regular expression or by a StringMatchingRule. The StringMatchingRule shall be provision whever possible. |

\* \* \* \* First change \* \* \* \*

##### 5.2.4.a Type: FqdnMatchingRule

Table 5.2.4.x-1: Definition of type FqdnMatchingRule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| fqdnRegexList | array(string) | O | 1..N | List of FQDN patterns, where each FQDN pattern is a regular expression according to the ECMA-262 dialect [17].(NOTE) |
| fqdnMatchingList | array(StringMatchingRule) | O | 1..N | List of FQDN patterns, where each FQDN pattern is described as a string match rule.(NOTE) |
| NOTE: When provisioning an FQDN pattern, the StringMatchingRule shall be preferred over regular expression and used whenever possible (i.e. if the pattern can be described by a string matching rule) to optimize the matching process and reduce the processing load, e.g. in the UPF or EASDF, since the use of regular expressions can be more computing intensive than using string matching rule. Either the fqdnRegexList or the fqdnMatchingList shall be present. |

\* \* \* \* Next change \* \* \* \*

##### 5.2.4.x Type: StringMatchingRule

Table 5.2.4.x-1: Definition of type StringMatchingRule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| stringMatchingConditions | array(StringMatchingCondition) | M | 1..N | Contains a list of conditions which shall be evaluated for string matching. |
| NOTE: The conditions in the stringMatchingConditions array shall be evaluated as "and" logical relationship. |

\* \* \* \* Next change \* \* \* \*

##### 5.2.4.y Type: StringMatchingCondition

Table 5.2.4.y-1: Definition of type StringMatchingCondition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| matchingString | string | M | 1 | Identifies the string against which the matching is performed. |
| matchingOperator | MatchingOperator | M | 1 | Identifies the matching operation. |

\* \* \* \* Next change \* \* \* \*

##### 5.2.3.x Enumeration: MatchingOperator

Table 5.2.3.x-1: Enumeration MatchingOperator

|  |  |  |
| --- | --- | --- |
| Enumeration value | Description  | Applicability |
| FULL\_MATCH | Indicates a full match between the string against which the matching applies and the provided matching string. |  |
| MATCH\_ALL | Indicate a match for any string |  |
| STARTS\_WITH | Indicates a match when the string against which the matching applies starts with the provided matching string (e.g. the string "smartmeter-01.company.com” matches the string "smartmeter-"). |  |
| NOT\_START\_WITH | Indicates a match when the string against which the matching applies not starts with the provided matching string (e.g. the string "smartmeter-01.company.com” matches the string “metersmart-"). |  |
| ENDS\_WITH | Indicates a match when the string against which the matching applies ends with the matching string (e.g. the string "somehost.company.com" matches the string "company.com"). |  |
| NOT\_END\_WITH | Indicates a match when the string against which the matching applies does not end with the matching string (e.g. the string "somehost.company.com" matches the string "company.se"). |  |
| CONTAINS | Indicates a match when the string against which the matching applies is contained within the matching string (e.g. the string "media.news.com" matches the string "media"). |  |
| NOT\_CONTAIN | Indicates a match when the string against which the matching applies is not contained within the matching string (e.g. the string "media.news.com" matches the string "aidem"). |  |

\* \* \* \* Next change \* \* \* \*

## A.2 Data related to Common Data Types

openapi: 3.0.0

info:

 version: '1.3.0-alpha.5'

 title: 'Common Data Types'

 description: |

 Common Data Types for Service Based Interfaces.

 © 2022, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

 All rights reserved.

externalDocs:

 description: 3GPP TS 29.571 Common Data Types for Service Based Interfaces, version 17.5.0

 url: 'https://www.3gpp.org/ftp/Specs/archive/29\_series/29.571/'

**...Skipped for clarity...**

 NullValue:

 enum:

 - null

 description: JSON's null value.

 MatchingOperator:

 anyOf:

 - type: string

 enum:

 - FULL\_MATCH

 - MTACH\_ALL

 - STARTS\_WITH

 - NOT\_START\_WITH

 - ENDS\_WITH

 - NOT\_END\_WITH

 - CONTAINS

 - NOT\_CONTAIN

 - type: string

 description: the matching operation.

**...Skipped for clarity...**

 TunnelAddress:

 description: Tunnel address

 type: object

 properties:

 ipv4Addr:

 $ref: '#/components/schemas/Ipv4Addr'

 ipv6Addr:

 $ref: '#/components/schemas/Ipv6Addr'

 portNumber:

 $ref: '#/components/schemas/Uinteger'

 required:

 - portNumber

 anyOf:

 - required: [ ipv4Addr ]

 - required: [ ipv6Addr ]

 FqdnMatchingRule:

 description: a list of FQDNs to be matched

 type: object

 not:

 required: [ fqdnMatchingList, fqdnRegexList ]

 properties:

 fqdnRegexList:

 type: array

 items:

 type: string

 minItems: 1

 fqdnMatchingList:

 type: array

 items:

 $ref: '#/components/schemas/StringMatchingRule'

 minItems: 1

 StringMatchingRule:

 description: A list of conditions for string matching

 type: object

 properties:

 stringMatchingConditions:

 type: array

 items:

 $ref: '#/components/schemas/StringMatchingCondition'

 StringMatchingCondition:

 description: A String with Matching Operator

 type: object

 properties:

 matchingString:

 type: string

 matchingOperator:

 $ref: '#/components/schemas/MatchingOperator'

\* \* \* \* End of changes \* \* \* \*