**3GPP TSG-CT WG4 Meeting #125****C4-244346**

**Hefei, P.R. China; 14th – 18th October 2024 *was* C4-244299**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.3* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **29.510** | **CR** | **1082** | **rev** | **1** | **Current version:** | **19.0.0** |  |
|  | | | | | | | | |
| *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:*** | Canary test procedure clarifications | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Ericsson | | | | | | | | | |
| ***Source to TSG:*** | CT4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | SBIProtoc19 | | | | |  | ***Date:*** | | | 2024-10-14 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | F |  | | | | | ***Release:*** | | | Rel-19 |
|  | *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 can be 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-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)  Rel-20 (Release 20)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | The procedure described in Annex D assumes that the SW upgrade also involves isolating the traffic and cleaning up the existing contexts/sessions in the NF Service Producer to be upgraded. However, there are cases in which the operator requires ISSU (In-Service-Software-Upgrade) and keep the existing contexts/sessions but at the same time steer canary traffic (e.g. for test UEs) towards the NF Service Producer after the SW upgrade (e.g. during canary testing for new features). | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | The sequence to introduce canary testing and at the same time keep the context (for in service SW upgrades) is clarified so that some actions are stated as optional or subject to operator’s policies. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | The normative Annex describes a procedure which does not cover the cases where operators require in-service SW upgrade. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | Annex D | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **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 does not introduce any changes in OpenAPI descriptions. | | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

\* \* \* First Change \* \* \* \*

Annex D (normative):  
Support of "Canary Release" testing in the NRF

This feature allows a network operator to deploy new software features in a Network Function (NF) in a controlled manner, by isolating the NF (Service) Instances that implement the new features, and steering solely a part of the traffic that, otherwise, would be sent towards such NF (Service) Instance producer.

This is achieved by means of the following mechanisms:

- Setting the NF (Service) Instance in Canary Release condition. This can be done by:

- setting the NFStatus, or NFServiceStatus, of the NF service producer, to the value "CANARY\_RELEASE", or

- setting the "canaryRelase" attribute to true in the NFProfile or NFService, while keeping the NF(Service)Status as "REGISTERED"

- Defining in the NFProfile or NFService of the NF service producer a set of selection conditions, that will be evaluated by an NF service consumer when attempting to select a candidate producer

The set of conditions may include, e.g.:

- The NF type of the consumer

- A feature number that is required by the consumer to select (and send traffic to) a producer

- A set of specific UEs (e.g. based on SUPI ranges, GPSI ranges, IMPU/IMPI ranges, list of PEIs)

- Any UE camping on a TAI within a set of TAI ranges

- A list of DNNs

In order to allow flexibility in the definition of the selection conditions, the above conditions can be combined by means of "and" / "or" logical operators.

EXAMPLE: An SMF is deployed with a new software feature, and the operator wishes to test it by carrying traffic to it with the following conditions:

- The SMF shall ony be selected by an AMF, a NEF, or an NWDAF

- The SMF shall only be selected by the AMF when:

- its selection requires the support of the "ATSSS" feature (which corresponds with feature number #2 in the "nsmf-pdusesssion" service)

- the UE belongs to a certain range of SUPIs

- the UE is camping on a TAI belonging to a range of TAIs

- When the SMF is selected by the NEF, it shall only do it for a certain DNN

- When the SMF is selected by the NWDAF, it shall onlly do it when the UE is camping on a TAI belonging to a range of TAIs

An example of the "selectionConditions" attribute could be as follows (note that there might be different logical expressions to encode the same logic):

"selectionConditions": {

"or": [

{

"and": [

{ "consumerNfType": [ "AMF" ] },

{ "serviceFeature": 2 },

{ "supiRange": { "start": "1234511111", "end": "1234599999" } },

{ "taiRange": {

"plmnId": { "mcc": "123", "mnc": "45" },

"tacRangeList": [

{ "start": "000011", "end": "0000ff" }

]

}

}

]

},

{

"and": [

{ "consumerNfType": [ "NEF" ] },

{ "dnnList": [ "internet.operator.com" ] }

]

},

{

"and": [

{ "consumerNfType": [ "NWDAF" ] },

{ "taiRange": {

"plmnId": { "mcc": "123", "mnc": "45" },

"tacRangeList": [

{ "start": "000011", "end": "000022" }

]

}

}

]

}

]

}

or, alternatively, with a more simplified encoding, based on the fact that the individual conditions (attributes) inside ConditionItem (see clause 6.1.6.2.124) are evaluated following the "AND" logical operator:

"selectionConditions": {

"or": [

{

"consumerNfType": [ "AMF" ],

"serviceFeature": 2,

"supiRange": { "start": "1234511111", "end": "1234599999" },

"taiRange": {

"plmnId": { "mcc": "123", "mnc": "45" },

"tacRangeList": [

{ "start": "000011", "end": "0000ff" }

]

}

},

{

"consumerNfType": [ "NEF" ],

"dnnList": [ "internet.operator.com" ]

},

{

"consumerNfType": [ "NWDAF" ],

"taiRange": {

"plmnId": { "mcc": "123", "mnc": "45" },

"tacRangeList": [

{ "start": "000011", "end": "000022" }

]

}

}

]

}

Once the producer has defined the selection conditions in its NFProfile, registered at the NRF, the sequence of steps for selection an NF after it is deployed with a "canary" software release, would be as follows:

1. Subject to operator’s policy, if the NF Service Instance to be software-upgraded is not to be kept operative, it changes its status to "UNDISCOVERABLE", and potential consumers are notified

2. If previous step was executed, NF Service Instance gets progressively emptied of existing sessions, until no sessions remain

3. If previous steps were executed, NF Service Instance may further change its status to "SUSPENDED", to ensure that no residual traffic is sent to the NF Service Instance, and potential consumers are notified

NOTE: In the scenario of in-service software upgrade, the operator does not empty the existing sessions in the NF Service Instance to be upgraded. The benefit of this is that traffic re-distribution is avoided when traffic is not matching the selection conditions.

4. NF Service Instance is software-upgraded

5. NF Service Instance with new software registers in NRF indicating its Canary Release condition, by setting its NF(Service)Status to "CANARY\_RELEASE" or by setting the "canaryRelease" attribute to true while keeping its NF(Service)Status as "REGISTERED", and includes the set of conditions for selection (e.g. a SUPI range)

6. Consumer NF sends discovery to NRF with usual discovery parameters and gets matching NF Instances containing NF Services with status "REGISTERED" or "CANARY\_RELEASE"

7. Consumer performs selection from the list of candidate NFs and, if the producer NF is in Canary Release condition, such NF shall only be selected if the selection conditions match (e.g. it shall only be selected if the SUPI matches the SUPI range indicated by the producer NF).

The selection of the candidate producers shall take into account the attributes of the discovered NFProfiles, and in addition, the consumer shall evaluate the attributes in the selection conditions, which shall take precedence over the attributes of the NFProfile of the producer, for those NF (Service) Instances in Canary Release condition.

If multiple candidate producers are available with NF (Service) status set to "REGISTERED" or "CANARY\_RELEASE", the consumer shall preferably select a producer in "CANARY\_RELEASE" status except if the "exclusiveCanaryReleaseSelection" attribute is present and set to true; in this case, the consumer shall only select producers in Canary Release condition (for which the selection conditions match), and shall not select those producers that are not in Canary Release condition.

NOTE: In the scenario of exclusive selection of Canary NF producers, the operator typically conducts very restricted Canary tests where the NF consumer only considers as valid candidate NF producer those instances in Canary condition, and discards all other NF producers (in non-Canary condition); the benefit of these tests is to check if the traffic case (for the selection conditions of the Canary nodes) actually end up in success or failure, rather than, e.g., ensuring that the traffic cases complete successfully after the consumer re-selects a non-Canary candidate producer. In these test scenarios, the operator need to ensure that the "exclusiveCanaryReleaseSelection" flag is consistently set in all the Canary NF producers; otherwise, the consumer can send traffic exclusively to Canary NF producers as long as there is one instance among the candidate producers having the "exclusiveCanaryReleaseSelection" set to true.

For the case of Indirect Communication, the selection of a candidate producer may be done by the SCP. In that case, the SCP needs to be able to evaluate the selection conditions for those producers in Canary Release condition. Since the SCP does not count with this information at its disposal (e.g. the different identities of the UE for which a service request is invoked via the SCP), it shall be provided by the consumer, e.g. by including the "3gpp-Sbi-Correlation-Info" HTTP header or by including the corresponding discovery headers ("3ggp-Sbi-Discovery") containing the UE identities involved in the specific traffic case.

In certain cases, the operator may want to maintain the NF Instance fully operative (so it keeps serving traffic to any consumer), at the same time as testing the new software features; in such case, the operator may deploy distinct service instances, some of which may keep the old software version and keep the "REGISTERED" NFServiceStatus, while other service instances may be deployed with the new software version, and set the NFServiceStatus to "CANARY\_RELEASE", to ensure that it is only selected by consumers when the desired conditions are met.

Another alternative approach to maintain the node fully operative may be achieved by setting the "canaryRelease" indication to true in the NFProfile or NFService of the NF (Service) instance, while keeping both the NFStatus and NFServiceStatus as "REGISTERED". This way, all consumers that do not support the Canary Release feature will keep discovering and potentially selecting such NF (Service) instance, so it will keep serving traffic normally for legacy consumers; however, if the consumer supports the Canary Release feature, it shall check the "canaryRelease" flag and, if set to true, it shall evaluate the selection conditions, similarly as if the NF(Service)Status would be set to "CANARY\_RELEASE".

An SCP or SEPP Network Entities may be set under Canary Release condition and define a set of selection conditions at NFProfile level to indicate to potential consumers that such Network Entity (SCP or SEPP) should only be selected if the selection conditions match.

Given that there are no services defined for an SCP, it is not possible to have an SCP "operative" (i.e. to be able to serve traffic to any consumer) when the status is set to "CANARY\_RELEASE" and the consumers might not support the "CANARY\_RELEASE" feature. However, it is possible to keep the SCP operative (and serve traffic to consumers not supporting such feature) by setting the "canaryRelease" attribute to true in the NFProfile of the SCP, and keep the NFStatus as "REGISTERED".

For a SEPP, the same considerations as with SCP apply, since there are no SEPP services per-se, in relation to the selection of SEPP instances, and the invocation of its N32-related message transfer capabilities.

However, a SEPP Network Entity may set the NFServiceStatus of a "nsepp-telescopic" service instance (while keeping the NFStatus of the SEPP set to "REGISTERED"). In this case, it is possible to have several instances of this service deployed, where some of them have the NFServiceStatus set to "REGISTERED" (and any consumer may select them) and others set to "CANARY\_RELEASE", so consumers should only select such service instance when the selection conditions match.

\* \* \* End of Change \* \* \* \*