**3GPP TSG-SA5 Meeting #134e *S5-206256rev3***

**e-meeting**

**Source: Ericsson**

**Title:** **Proposal on clarification of ServiceProfile representations**

**Document for: Endorsement**

**Agenda Item: 6.3**

# 1 Decision/action requested

**The group is asked to discuss and agree on the proposal.**

# 2 References

[1] 3GPP TS 28.531 V16.7.0 Management and orchestration; Provisioning;

[2] 3GPP TS 28.541 V16.6.0 Management and orchestration; 5G Network Resource Model (NRM);

[3] 3GPP TS 28.530 V16.3.0 Management and orchestration; Concepts, use cases and requirements;

[4] S5-205405 Proposal on updates to network slice model and procedures (Endorsed SA5 #133e)

[5] [S5-205268](https://www.3gpp.org/ftp/TSG_SA/WG5_TM/TSGS5_133e/Docs/S5-205268.zip) Proposal on updates to network slice model and procedures (original proposal, become endorsed S5-205405)

# 3 Rationale

In last SA5 #133e meeting, a discussion paper named, “Proposal on updates to network slice model and procedures” were proposed and discussed [5]. This discussion paper addresses the questions raised in the e-mail tread when it comes to what the ServiceProfile represents, the required updates to operations/procedures and around capabilities. We think that it is important that SA5 can come to a common understanding around the network slice “concept” defined.

The proposal 5 and 6 in the discussion paper to SA5 #133e meeting [5] around the NetworkSlice as well as ServiceProfile are made extendable, that we see as important part of the network slice concept, is not part of this paper (separate contribution).

**Background:**

**Observation 1:** The ServiceProfile and the SliceProfile should best be understood as input “requirements” from Consumer to the Producer, i.e. what the Communication Service (CS) or a NetworkSlice-as-a-Service (NSaaS) needs/requests. Example of requirements can be coverage, latency etc.

**Observation 2:** For the Provider to understand that a network slice represent a Communication Service (CS) or a NetworkSlice-as-a-Service (NSaaS), an indication is needed. This information may e.g. be used by the Provider to deduce what capabiltities or services shall be exposed to the Consumer (TBD). If the slice represents NSaaS, a new dedicated NetworkSlice instance shall be created/allocated.

**Observation 3:** There are in SA5 no common understanding around meaning of the optional attribute resourceSharingLevel in ServiceProfile. It can not be agreed in release-16 that this attribute is related to input requirement on logical resources. But there is an understanding in SA5 that when resourceSharingLevel = “non-shared” the requested service (CS or NSaaS) in the ServiceProfile must be the the only service in the NetworkSlice instance (only one service in the NetworkSlice instance).

**Observation 4**: Based on the ServiceProfile provided as input in the *allocateNsi* procedure the Producer allocates a NetworkSlice instance identified by a network slice identity nsId. If the ServiceProfile.resourceSharingLevel is “shared”, there is no information to the Consumer whether the slice instance is shared with other communications services or not. In order to allow the Consumer to also remove the communications service from the network slice instance without affecting other communications services, an identity if the service profile requirements is, to be used in the deallocateNsi procedure. Similarly, an identity of the slice profile requirements is added as parameter in the deallocateNssi procedure. It is understood that the Consumer via NRM can get knowledge to the relation between serviceProfileId and nsId.

**Observation 5**: The Provider should be able to assign the S-NSSAI associated with each ServiceProfile in a NetworkSlice. This is valid when the ServiceProfile represents a Communication Service (CS), or a NetworkSlice-as-a-Service (NSaaS). In case of NSaaS it, there might be a UC there the Consumer need to assign/coordinate the use of S-NSSAI, when the two operators are involved in the e2e service, a rel-17 study item.

Further, the S-NSSAI is defined in a PLMN, it is important to get this relation correct into the ServiceProfile in slice NRM. The PLMNInfo <<datatype>> is used in NR NRM to cover this relation.

**Based on the above observations we make the following proposals:**

**Proposal 1:** Based on observation 1, the specifications should make it unequivocally clear that the ServiceProfile represents the service requirements on the NetworkSlice instance for a particular requested service (by a customer). It does not represent the actual NetworkSlice instance deployed and its capabilities. With this clarification, there can be a 1: n relation between NetworkSlice and ServiceProfile, as one NetworkSlice can carry more than one service as long as these do not impose conflicting requirements. Similarly, the SliceProfile represents the requirements for the NetworkSliceSubnet.

In the specific case the NetworkSlice itself is the service, as defined for NSaaS [3], there will only be one ServiceProfile defining the input requirements for the NetworkSlice. Thus, for the scenario when the NetworkSlice itself is the service, there is a 1:1 relation between ServiceProfile and NetworkSlice.

**Proposal 3:** Based on observation 3, it is proposed to rename and redefine the attribute ServiceProfile.resourceSharingLevel to not cover any resource aspects. Proposed new name is serviceProfile.**networkSlice**SharingLevel, see detailed proposal, in section 4, “Proposal 3”

**Proposal 4**: Based on observation 3, the allocateNsi operation/procedure needs to be updated to reflect that a ServiceProfile represents the service requirements on the NetworkSlice instance and that a new network slice instance is always created if ***networkSlic****eSharingLevel* is equal to “non-shared”.

**Proposal 5:** Based on observation 3, the allocateNssi operation/procedure needs to be updated to reflect that a SliceProfile represents the service requirements on the NetworkSliceSubnet instance and that a new slice subnet instance is always created if ***networkSliceSubnet****SharingLevel* is equal to “non-shared” if sufficient resouces are available.

**Proposal 6**: Based on observation 4, the deallocateNSi procedure is updated to take serviceProfileId and nsId as input parameters. The Provider removes the ServiceProfile corresponsing to the serviceProfileId in the network slice instance identified by nsId. When the last ServiceProfile is removed from a network slice instance, the Provider may decide to remove the network slice instance. If serviceProfileId is omitted in the deallocateNsi request, the Provider removes the network slice instance identified by nsId*.* Similarly, the sliceProfileId is added to nssId as parameter in the deallocateNssi procedure.

**Proposal 7:** Based on observation 5, and the S-NSSAI configuration in ServiceProfile, we propose:

* to make S-NSSAI(s) assignment/configuration in ServiceProfile, done by the NetworkSlice instance Producer, as a result of a successful allocateNsi operation.
* to replace existing sNSSAIList and PLMNId attributes with PLMNInfoList instead to always get the correct S-NSSAI and PLMNId relation. (alignment with NR NRM)

# 4 Detailed proposal

Based on the observations above, we ask for endorsement of proposal 3, 4, 5, 6. 7, to clarify the representation of the serviceProfile and updates to operations/procedures:







**Proposal 3**: Rename and redefine the attribute ServiceProfile.resourceSharingLevel to ServiceProfile.**networkSlice**SharingLevel to not cover resource aspects.. (update to serviceProfile and to attribute definition in 28.541 [2])

| serviceProfile.**networkSlice**SharingLevel | **An attribute specifies whether the service (CS) defined in a serviceProfile can be shared with other services in a networkSlice instance.**  **If this attribute is equal:**  **“non-shared”: then a new networkSlice instance must always be created.**  **“shared”: then the service may be together (shared) with other service(s) in a networkSlice instance.**  allowedValues: shared, non-shared. | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  allowedValues: Yes  isNullable: True |
| --- | --- | --- |

**Proposal 4**: Update description of the allocateNsi and deallocate procedures in 28.531 [1]. Note that the text in “grey” below is depending on endorsement of proposal 3.

* Update subclause 6.5.1.1 (allocateNsi), by adding below text in bold.

This operation is invoked by allocateNsi operation service consumer to request the provider to allocate a network slice instance to satisfy network slice related requirements. The provider may create a new NSI or using existing NSI to satisfy the request. **The requirements in the request are compared/matched against the actual capabilitites of all candidate NSIs. If an NSI can be found e.g. with the right coverage and with good enough latency, it is eligible for allocation. In case not, or if *networkSliceSharingLevel* is equal to “non-shared”, a new NSI is created with capabilities to host the service, given that required NSSIs can be created.**

* Update subclause 6.5.3.1 (deallocateNsi), by adding below text in bold.

This operation is invoked by deallocateNsi operation service consumer to request the provider to deallocate **a service in an NSI or** a network slice instance since the NSI is no longer needed for the consumer. The provider may terminate the requested NSI or modify the requested NSI without termination to satisfy the request.

**Proposal 5**: Update description of the allocateNssi and deallolcateNssi procedures in 28.531 [1]. Note that the text in “grey” below is depending on endorsement of proposal 3.

* Update subclause 6.5.2.1(allocateNssi), by adding below text in bold.

This operation is invoked by allocateNssi operation service consumer to request the provider to allocate a network slice subnet instance to satisfy the network slice subnet related requirements. The provider may create a new NSSI or using existing NSSI to satisfy the request. **The requirements in the request are compared/matched against the actual capabilitites of all candidate NSSIs. If an NSSI can be found e.g. with the right coverage and with good enough latency, it is eligible for allocation. In case not, or if *networkSliceSubnetSharingLevel* is equal to “non-shared”, a new NSSI is created with capabilities to host the service, if enough resoures are available.**

* Update subclause 6.5.4.1, by adding below text in bold.

This operation is invoked by deallocateNssi operation service consumer to request the provider to deallocate **a slice in the NSSI or** a network slice subnet instance since the NSSI is no longer needed for the consumer. The provider may terminate the requested NSSI or modify the requested NSSI without termination to satisfy the request.

**Proposal 6**: Update the procedures deallocateNsi and deallocateNSsi in 28.531 [1] according to below.



* Add serviceProfileId as input to the deallocateNsi procedure in subclause 6.5.3.2.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter Name | Support Qualifier | Information Type / Legal Values | Comment |
| serviceProfileId | O | An attribute uniquely identifies the service profile in an NSI. | It specifies the unifique identifier of the service profile in the NSI which is to be deallocated. If omitted, all service profiles in the NSI are in scope and the Provider may decide to remove the NSI. |



* Add serviceProfileId as input to the deallocateNssi procedure in subclause 6.5.4.2.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter Name | Support Qualifier | Information Type / Legal Values | Comment |
| sliceProfileId | O | An attribute uniquely identifies the slice profile in an NSI. | It specifies the unifique identifier of the slice profile in the NSSI which is to be deallocated. If omitted, all slice profiles in the NSSI are in scope and the Provider may decide to remove the NSSI. |

**Proposal 7**: Replace the attributes pLMNIdList and sNSSAIList with PLMNInfoList in the ServiceProfile in 28.541 [2].

* Modify ServiceProfile in subclause 6.3.3.2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| serviceProfileId | M | T | F | T | T |
| ~~sNSSAIList~~ | ~~M~~ | ~~T~~ | ~~T~~ | ~~F~~ | ~~T~~ |
| ~~pLMNIdList~~ | ~~M~~ | ~~T~~ | ~~T~~ | ~~F~~ | ~~T~~ |
| **pLMNInfoList** | **M** | **T** | **F** | **F** | **T** |

* Modify SliceProfile in subclause 6.3.4.2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| sliceProfileId | M | T | F | T | T |
| ~~sNSSAIList~~ | ~~M~~ | ~~T~~ | ~~T~~ | ~~F~~ | ~~T~~ |
| ~~pLMNIdList~~ | ~~M~~ | ~~T~~ | ~~T~~ | ~~F~~ | ~~T~~ |
| **pLMNInfoList** | **M** | **T** | **T** | **F** | **T** |

* Add ServiceProfile.pLMNInfoList and SliceProfile.pLMNInfoList attribute properties into subclause 6.4.1.

| Attribute Name | Documentation and Allowed Values | Properties |
| --- | --- | --- |
| ServiceProfile.pLMNInfoList | It defines which PLMN and S-NSSAI combinations that are served by the ServiceProfile in case of network slicing feature is supported.  allowedValues: Not applicable. | type: PLMNInfo  multiplicity: 1..\*  isOrdered: N/A  isUnique: True  defaultValue: None  isNullable: False |
| SliceProfile.pLMNInfoList | It defines which PLMN and S-NSSAI combinations that are served by the SliceProfile in case of network slicing feature is supported.  allowedValues: Not applicable. | type: PLMNInfo  multiplicity: 1..\*  isOrdered: N/A  isUnique: True  defaultValue: None  isNullable: False |







# 5 Conclusion

We ask for endorsement of the proposals 3, 4, 5, 6, 7 above.

Once the endorsement is agreed, the required CRs towards 28.541, 28.530, 28.531 will be produced for the SA5 #135e meeting.