**3GPP TSG-SA5 Meeting #133e *S5-205268rev2***

**e-meeting 12th - 21st October 2020**

**Source: Ericsson**

**Title: Proposal on updates to network slice model and procedures**

**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;

# 3 Rationale

This paper discuss some of the shortcomings of the current 3GPP SA5 specifications for network slice creation/orchestration and suggests some steps to remedy these. To be able to fully automate network slice creation and with interoperability, the specifications must expose a much higher degree of flexibility, i.e to not only have characteristics of the network slice as input. A set of proposals will be discussed to make the specifications more adaptable and usable in real deployments of network slicing.

 Note, that the proposals in this discussion paper does not add any changes to the Rel-16 slice definition [1], nor to the use of GSMA attributes. This discussion paper focus on the network slice orchestration but is also valid to network slice subnet.

**Background:**

Network slices can in reality come in many shapes and forms. There are many potential use cases and potential business offerings leveraging slicing, some known and likely many unknown. Each such case can also be realized in multiple ways and need to be *adapted to considerations related to deployment and operator processes.* Due to this, the space of possible network slice types (with different functionality, different functional distribution and so on) quickly reaches the 1000s if not more. In reality there will be an evolutionary downselection to those that will really drive the business, but it is impossible to upfront do this selection, both in standardization as well as in realization, this would require a “crystal ball”.

Due to this, the only way to ensure that dead ends are avoided, is to equip the standards and solutions with flexibility, viewing the design/construction of the network slice types as a slice design process, as opposed to hard coding everything in standardization or in the vendors implementation.

Current SA5 specifications have focused on the characteristics of a network slice, defined in ServiceProfile and its mapping to a SliceProfile. Those characteristics aspects are essential, and thus nothing wrong per se. However, a real deployment of network slice management with automated network slice creation and with interoperability between Consumer and Producer, there is also a need to consider other aspects not yet taken into consideration in SA5, which is the subject of this discussion paper.

**Observation 1:** The ServiceProfile (and in similar ways the SliceProfile) should best be understood as “requirements”, i.e. what the service needs/requests. The service needs a particular coverage, latency etc. When selecting a network slice however, these needs must be matched against the actual capabilities of a network slice. Not all slices will be able to deliver low latency, slices will have different coverage etc. An actual network slice is defined and constrained by one set of capabilities. The current specifications lack clarity on how to semantically handle the ServiceProfile, i.e. is it just defining requirements, or is it also defining the actual network slice? Furthermore, no explicit modelling exists for the actual capabilitites of the NetworkSlice instance. The current ambiguity of the ServiceProfile leads to unnecessary discussions about “cardinality”.

**Observation 2:** The currently specified ServiceProfile parameters can only be interpreted as the “best current guess” of what would actually be required in the field to make network slicing work. To avoid long lead times of changing the standards, the specifications need to be equipped with more flexibility, i.e. the ability to add new parameters as needed together with their associated meta data. The current specifications are too rigidly bound to the specified parameters.

**Observation 3:** Real life deployments require other parameters to be exchanged related to the NetworkSlice instance in addition to the network slice capabilities. It ranges from basic things such as peer IP addresses to operational parameters, deployment specifics and policies for usage of resources. To cater for this, while keeping interoperability, there is a need to be able to extend the information models used across the interfaces “in the field” as opposed to standardizing everything (which is too slow). There is therefore a need to make the operations more flexible and model driven as opposed to rigidly specified.

**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 that are put on the NetworkSlice instance for a particular requested service (by a customer). It does not represent the actual NetworkSlice instance 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.

**Proposal 2:** Based on observation 1, the NetworkSlice gets a set of attributes representing the actual capabilitites of an allocated NetworkSlice instance, for instance the actual coverage of the NetworkSlice instance, or the minimum latency that it is capable of delivering. These parameters can be hosted by a <<dataType>>, e.g. called NetworkSliceCapabilities (see Figure below).

**Proposal 3:** Based on observation 1, the procedures of TS 28.531 are updated to clarify that, as part of e.g. “AllocateNsi”, the provided ServiceProfile (the requirements) are to be compared/matched against the actual capabilitites (NetworkSliceCapabilities) of all the candidate NetworkSlice instances. If a NetworkSlice instance can be found e.g. with the right coverage and with good enough latency, it is eligible for allocation. In case not, a new NetworkSlice instance must be created that has the right capabilities to host the service.

**Proposal 4:** Based on observation 1, an evaluation needs to be done about what attributes should be part of ServiceProfile, NetworkSlice or both and their corresponding definitions.

**Proposal 5:** Based on observations 2 and 3, both the NetworkSlice as well as ServiceProfile are made extendable in the sense that new attributes can be added. The operations (AllocateNsi) must be extendable to allow other parameters than those standardized to be managed in a standardized way.

**Proposal 6:** Related to observation 3 it is proposed that the meta data describing the additional data for ServiceProfile and NetworkSlice is made discoverable by the Consumer adding a new AdditionalDataSpec IOC (see Figure below). That opens for extendibility, yet in a way where multi-vendor interworking is fully supported.



Figure: Shows the proposed addition with respect to networkSliceCapabilities and extendabilities of additionalServiceProfileDataList and additionalNetworkSliceDataList, described in “Detailed Proposal” below.

Note: Here we focus on network slice level, but similar changes shall be made on network slice subnet level.

# 4 Detailed proposal

Extensions to the NRM in TS 28.541 is elaborated below:



4.1 NetworkSlice

4.1.1 Definition

This IOC represents the properties of a network slice instance in a 5G network. For more information about the network slice instance, see 3GPP TS 28.531 [1].

4.1.2 Attributes

The NetworkSlice IOC includes attributes inherited from SubNetwork IOC (defined in TS 28.622) and the following attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute name** | **Support Qualifier** | **isReadable** | **isWritable** | **isInvariant** | **isNotifyable** |
| operationalState | M | T | F | F | T |
| administrativeState | M | T | T | F | T |
| serviceProfileList | M | T | T | F | T |
| networkSliceCapabilities | M | T | F | T | T |
| additionalNetworkSliceDataList | O | T | T | F | T |
|  |  |  |  |  |  |
| **Attribute related to role** |  |  |  |  |  |
| networkSliceSubnetRef | M | T | F | F | T |
| additionalDataSpecRef | O | T | F | T | T |

4.1.3 Attribute constraints

None.

**4.1.4 Attribute properties**

| Attribute Name | Documentation and Allowed Values | Properties |
| --- | --- | --- |
| additionalNetworkSliceDataList | This parameter specifies the list of additional data attributes and values defined by the referred AdditionalDataSpec IOC for the network slice instance. | type: AdditionalDataValue multiplicity: 1..NisOrdered: FalseisUnique: N/AdefaultValue: NoneallowedValues: N/AisNullable: True |
| additionalDataSpecRef | This holds a DN of an AdditionalDataSpec instance defining meta data for the additionalNetworkSliceDataList in the NetworkSlice and additionalServiceProfileDataLists in the associated ServiceProfiles.  | type: DNmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |

### 4.2 NetworkSliceCapabilities <<dataType>>

#### 4.2.1 Definition

This datatype represents the capabilitites of a NetworkSlice instance in a 5G network. For more information about the NetworkSlice instance, see 3GPP TS 28.531 [1].

#### 4.2.2 Attributes

The NetworkSliceCapabilities includes attributes defining constraints to the associated ServiceProfile(s):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| latency | M | T | F | F | T |
| … |  |  |  |  |  |
| **Attribute related to role** |  |  |  |  |  |

Note1: The attribute latency here defines the minimum latency supported by the network slice, in contrast to latency as part of the ServiceProfile, which defines the requirement from the associated service. The latency of the NetworkSlice instance satisfies the latency of the ServiceProfile(s) but is not necessarily equal.

Note2: The latency attribute is an **example** given here, the remaining attributes to define the network slice capabilities are proposed to be further discussed.

#### 4.2.3 Attribute constraints

None.

**4.2.4 Attribute properties**

| Attribute Name | Documentation and Allowed Values | Properties |
| --- | --- | --- |
| latency | This attribute defines the minimum latency supported by the NetworkSlice. It specifies the packet transmission latency (millisecond) through the RAN, CN, and TN part of 5G network and is used to evaluate utilization performance of the end-to-end network slice instance. See clause 6.3.1 of 28.554. | type: Integermultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneallowedValues: N/AisNullable: False |

### 4.3 ServiceProfile <<dataType>>

#### 4.3.1 Definition (from [2])

This data type represents the properties of network slice related requirement that should be supported by the network slice instance in 5G network. The network slice can be tailored based on the specific requirements adhered to SLA agreed between Network Slice Customer (NSC) and Network Slice Provider (NSP), see clause 2 of [50]. A network slicing provider may add additional requirements not directly derived from SLA’s, associated to the provider internal [business] goals. The GST defined by GSMA (see [50]) and the service performance requirements defined in 3GPP TS 22.261 [28] and TS 22.104 [51] are all considered as input for the network slice related requirements.

#### 4.3.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 |
| maxNumberofUEs | O | T | T | F | T |
| coverageArea | O | T | T | F | T |
| latency | O | T | T | F | T |
| uEMobilityLevel | O | T | T | F | T |
| resourceSharingLevel | O | T | T | F | T |
| sST | M | T | T | F | T |
| availability | O | T | T | F | T |
| delayTolerance | O | T | T | F | T |
| deterministicComm | O | T | T | F | T |
| dLThptPerSlice | O | T | T | F | T |
| dLThptPerUE | O | T | T | F | T |
| uLThptPerSlic | O | T | T | F | T |
| uLThptPerUE | O | T | T | F | T |
| maxPktSize | O | T | T | F | T |
| maxNumberofConns | O | T | T | F | T |
| kPIMonitoring | O | T | T | F | T |
| supportedAccessTech | O | T | T | F | T |
| userMgmtOpen | O | T | T | F | T |
| v2XCommModels | O | T | T | F | T |
| termDensity | O | T | T | F | T |
| activityFactor | O | T | T | F | T |
| uESpeed | O | T | T | F | T |
| jitter | O | T | T | F | T |
| survivalTime | O | T | T | F | T |
| reliability | O | T | T | F | T |
| additionalServiceProfileDataList | O | T | T | F | T |

#### 4.3.3 Attribute constraints

None.

#### 4.3.4 Attribute properties

| Attribute Name | Documentation and Allowed Values | Properties |
| --- | --- | --- |
| additionalServiceProfileDataList | This parameter specifies the list of additional data attributes and values defined by the AdditionalDataSpec IOC referred to by the associated NetworkSlice instance. | type: AdditionalDataValue multiplicity: 1..NisOrdered: FalseisUnique: N/AdefaultValue: NoneallowedValues: N/AisNullable: True |

### 4.4 AdditionalDataValue <<dataType>>

#### 4.4.1 Definition

This datatype defines the key value pair of additional attributes defined by the AdditionalDataSpec instance referred to by a NetworkSlice instance or a ServiceProfile instance.

#### 4.4.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| key | M | T | T | F | T |
| value | M | T | T | F | T |

#### 4.4.3 Attribute properties

| Attribute Name | Documentation and Allowed Values | Properties |
| --- | --- | --- |
| key | This attribute corresponds to an attributeName in the the AdditionalDataSpec instance referred to by a NetworkSlice instance or a ServiceProfile instance | type: string multiplicity: 1isOrdered: FalseisUnique: N/AdefaultValue: NoneallowedValues: N/AisNullable: True |
| value | This attribute corresponds to the value of the key. The data type of the value corresponds to the attributeType, in the the AdditionalDataSpec instance referred to by a NetworkSlice instance or a ServiceProfile instance, associated with the attributeName matching the key. | type: <given by attributeType in referred AdditionalDataSpec> multiplicity: 1isOrdered: FalseisUnique: N/AdefaultValue: NoneallowedValues: N/AisNullable: True |

### 4.5 AdditionalDataSpec <IOC>

#### 4.5.1 Definition

This IOC defines the meta data related to the networkSliceDataSpecList associated with a NetworkSlice instance and to the serviceProfileDataSpecList associated with the ServiceProfile(s) of the NetworkSlice instance.

#### 4.5.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| networkSliceDataSpecList | M | T | F | T | T |
| serviceProfileDataSpecList | M | T | F | T | T |
| **Attribute related to role** |  |  |  |  |  |

#### 4.5.3 Attribute properties

| Attribute Name | Documentation and Allowed Values | Properties |
| --- | --- | --- |
| networkSliceDataSpecList | This parameter specifies the list of additional data attributes defined by the referred AdditionalDataSpec IOC. | type: AdditionalDatamultiplicity: 1..NisOrdered: TrueisUnique: N/AdefaultValue: NoneallowedValues: N/AisNullable: True |
| serviceProfileDataSpecList | This parameter specifies the list of additional data attributes defined by the referred AdditionalDataSpec IOC. | type: AdditionalDatamultiplicity: 1..NisOrdered: TrueisUnique: N/AdefaultValue: NoneallowedValues: N/AisNullable: True |

### 4.6 AdditionalData <<dataType>>

#### 4.6.1 Definition

This AdditionalData <<dataType>> represents an attribute and its meta data in the AdditionalDataSpec IOC.

#### 4.6.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| attributeName | M | T | F | T | T |
| attributeMandatory | M | T | T | T | T |
| attributeType | M | T | T | T | T |
| attributeDefault | O | T | T | T | T |
| attributeMinValue | O | T | T | T | T |
| attributeMaxValue | O | T | T | T | T |
| attributeUnit | M | T | T | T | T |
| attributeDescription | O | T | T | T | T |
| **Attribute related to role** |  |  |  |  |  |

Note: The AdditionalData <<datatype>> includes attributes defining meta data associated with an attribute in the additionalNetworkSliceDataList of a NetworkSlice instance or associated with an attribute in the additionalServiceProfileDataList of a ServiceProfile.

| Attribute Name | Documentation and Allowed Values | Properties |
| --- | --- | --- |
| attributeName | This parameter defines the name of the attribute | type: stringmultiplicity: 1isOrdered: FalseisUnique: N/AdefaultValue: NoneallowedValues: N/AisNullable: False |
| attributeMandatory | Indicates whether a parameter is mandatory | type: booleanmultiplicity: 1isOrdered: FalseisUnique: N/AdefaultValue: FalseallowedValues: True/FalseisNullable: False |
| attributeType | This parameter defines the data type of the attribute, e.g. integer, float, boolean, string. | type: stringmultiplicity: 1isOrdered: FalseisUnique: N/AdefaultValue: NoneallowedValues: N/AisNullable: False |
| attributeDefault | This parameter defines the default value of the attribute | type: stringmultiplicity: 1isOrdered: FalseisUnique: N/AdefaultValue: NoneallowedValues: N/AisNullable: True |
| attributeMinValue | This parameter defines the minimum value allowed for the attribute | type: stringmultiplicity: 1isOrdered: FalseisUnique: N/AdefaultValue: NoneallowedValues: N/AisNullable: True |
| attributeMaxValue | This parameter defines the maximum value allowed for the attribute | type: stringmultiplicity: 1isOrdered: FalseisUnique: N/AdefaultValue: NoneallowedValues: N/AisNullable: True |
| attributeUnit | This parameter defines the unit of the attribute | type: stringmultiplicity: 1isOrdered: FalseisUnique: N/AdefaultValue: NoneallowedValues: N/AisNullable: False |
| attributeDescription | This parameter describes the attribute | type: stringmultiplicity: 1isOrdered: FalseisUnique: N/AdefaultValue: NoneallowedValues: N/AisNullable: True |

Extensions to the procedures in TS 28.531 is elaborated below:

### 4.7 AllocateNsi operation

#### 4.7.1 Description

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 NetworkSlice instance or using existing NetworkSlice instance to satisfy the request.

#### 4.7.2 Input parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter Name | Support Qualifier | Information Type / Legal Values | Comment |
| serviceProfile | M | LIST OF SEQUENCE< attribute name, attribute value> | This parameter specifies the network slice related requirements defined in ServiceProfile in Clause 6.3.3 in TS 28.541 [6]. |
| additionalServiceProfileDataList | O | LIST OF SEQUENCE< attribute name, attribute value> | This parameter specifies additional requirements related to the ServiceProfile  |
| additionalNetworkSliceDataList | O | LIST OF SEQUENCE< attribute name, attribute value> | This parameter specifies additional data related to the NetworkSlice  |

#### 4.7.3 Output parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter name | Support Qualifier | Matching Information / Legal Values | Comment |
|  |  |  |  |
| serviceProfile | M | LIST OF SEQUENCE< attribute name, attribute value> | This parameter specifies the network slice related requirements which has been allocated and the actual value assigned to each. |
| additionalServiceProfileDataList | O | LIST OF SEQUENCE< attribute name, attribute value> | This parameter specifies additional requirements related to the ServiceProfile which has been allocated and the actual value assigned to each. |
| additionalNetworkSliceDataList | O | LIST OF SEQUENCE< attribute name, attribute value> | This parameter specifies additional data related to the NetworkSlice which has been allocated and the actual value assigned to each. |
| status | M | ENUM (OperationSucceeded, OperationFailed) | An operation may fail because of a specified or unspecified reason. |
| nSId | M | An attribute uniquely identifies the network slice instance. | It specifies the unifique identifier of the NSI which has been allocated. |

# 5 Summary of proposals

Based on the observations and proposals in we ask for endorsement of the following:

First Part: Around adding NetworkSlice Capabilities:

1. From proposal1: that the ServiceProfile represents the service requirements that are put on the NetworkSlice instance for a particular requested service (by a customer).
2. From proposal2: that the NetworkSlice gets a set of attributes representing the actual capabilitites of a NetworkSlice instance. The capabilitites attributes to be placed in NetworkSliceCapabilities <<dataType>>.
3. From proposal3: the “allocateNsi” procedures of TS 28.531 are updated to clarify that, as part of e.g. “allocateNsi”, the provided ServiceProfile (the requirements) are to be compared/matched against the actual capabilitites (NetworkSliceCapabilities) of all the candidate NetworkSlice instances.

Second Part: Extendabilities of ServiceProfile and NetworkSlice:

1. From proposal5:
	* The NetworkSlice as well as ServiceProfile are made extendable in the sense that new additional attributes can be added to those.
	* The operations (allocateNsi) of TS 28.531 are updated to allow for the new additional attributes to be used as input parameters to allocateNsi.
2. From proposal6: It is proposed that the meta data describing the additional data for ServiceProfile and NetworkSlice are kept in a new AdditionalDataSpec IOC

The S5-205261 (Rel-16) and S5-205262 (Rel-17) are the stage 2 CRs to TS 28.541 and S5-205265 (Rel-16) is the stage 2 CR to TS 28.531.

# 6 Annex around the need for extendibility

In the SA5 #133e conference call this proposal was discussed, and a number of questions was reised. In this Annex, we tries to describe from a more “high level” description why this extendabilities are needed, to achive our goal described in this discussion paper: *To be able to fully automate network slice creation and with interoperability, the specifications must expose a much higher degree of flexibility, i.e to not only have characteristics of the network slice as input.*

An network slice can **not only** be created or compared/matched based on the customers service requirements in ServiceProfile, that holds the input service requirement for a wanted service. As highlighted in a number of specifications (example 28.541 6.6.3) the Producer of the network slice has sertain capabilities/requirements when it comes to its possibility to create a network slice with specific service characteristics. For a Producer to be able to create (allocate) a service to a network slice, he needs to have **both** service input requirements and those additional requirement/data from the Consumer.

Below a quick sketch to further explain our “Extendability” proposal in this paper. As seen from the sketch the Consumer first (1) learns parameters and types in the AdditionalDataSpec, from the Producer. Then (2) the Consumer populates and includes those additional attributes in the allocateNsi request together with service input requirement.

If the “format” how to exchange additional data is defined in slice NRM, Consumers from deferent vendors (x, y, x..) can get the knowledge over what additional data parameters that can/needs to be configured towards Producer (vendor x).



Analogy with ETSI:

In fact, our needed functionalities for network slice allocation have similarities/analogy with how ETSI deals with similar matters of interoperability. The difference really lies in the fact that ETSI relies on shared descriptors (VNFD, NSD), while our propose use an AdditionalDataSpec IOC to achieve the same thing.

