A Holistic, Innovative Framework for the Design, Development and Orchestration of 5G-ready Applications and Network Services over Sliced Programmable Infrastructure

Co-funded by the Horizon 2020 Framework Programme of the European Union

DELIVERABLE D1.4

NETWORK-AWARE APPLICATION GRAPH METAMODEL (NETWORK SLICE INTENT AND SLICE INSTANCE METAMODEL)

Due Date of Delivery:	M9 *Mx* (28/02/2018 *dd/mm/yyyy*)
Actual Date of Delivery:	5/3/2018 *dd/mm/yyyy*
Workpackage:	WP1 – MATILDA Reference Architecture, Conceptualization and Use Cases
Type of the Deliverable:	OTHER (XSD schema & documentation)
Dissemination level:	PU
Editors:	UBITECH, CNIT, UNIVBRIS, ININ, NCSRD
Version:	1.0

Call:
H2020-ICT-2016-2

Type of Action:
IA

Project Acronym:
MATILDA

Project ID:
761898

Duration:
30 months

Start Date:
01/06/2017

Project Coordinator:
Name: Franco Davoli
Phone: +39 010 353 2732
Fax: +39 010 353 2154
e-mail: franco.davoli@cnit.it

Technical Coordinator
Name: Panagiotis Gouvas
Phone: +30 216 5000 503
Fax: +30 216 5000 599
e-mail: pgouvas@ubitech.eu

List of the Authors	
CNIT	Consorzio Nazionale Interuniversitario per le Telecomunicazioni
Franco Davoli, Roberto Bruschi	
UBITECH	GIOUMPITEK Meleti Schediasmos Ylopoiisi kai Polisi Ergon Pliroforikis EPE
Panagiotis Gouvas, Anastasios Zafeiropoulos, Eleni Fotopoulou, Thanos Xirofotos	
UNIVBRIS	University of Bristol
Anderson Bravalheri	
ININ	Internet Institute, Communications Solutions and Consulting Ltd
Janez Sterle, Luka Koršič	
NCSRD	National Center for Scientific Research “DEMOKRITOS”
Eleni Trouva	

Disclaimer

The information, documentation and figures available in this deliverable are written by the MATILDA Consortium partners under EC co-financing (project H2020-ICT-761898) and do not necessarily reflect the view of the European Commission.

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The reader uses the information at his/her sole risk and liability.

Copyright

The MATILDA Consortium consists of:

CONSORZIO NAZIONALE INTERUNIVERSITARIO PER LE TELECOMUNICAZIONI

ATOS SPAIN SA (ATOS)

ERICSSON TELECOMUNICAZIONI (ERICSSON)

INTRASOFT INTERNATIONAL SA (INTRA)

COSMOTE KINITES TILEPIKOINONIES AE (COSM)

ORANGE ROMANIA SA (ORO)

EXXPERTSYSTEMS GMBH (EXXPERT)

GIOUMPITEK MELETI SCHEDIASMOS YLOPOIISI KAI POLISI ERGON PLIROFORIKIS ETAIREIA PERIORISMENIS EFTHYNIS (UBITECH)

INTERNET INSTITUTE, COMMUNICATIONS SOLUTIONS AND CONSULTING LTD (ININ)

INCELLIGENT IDIOTIKI KEFALAIOUCHIKI ETAIREIA (INC)

SUITE5 DATA INTELLIGENCE SOLUTIONS LIMITED (SUIT5)

NATIONAL CENTER FOR SCIENTIFIC RESEARCH "DEMOKRITOS" (NCSRD)

UNIVERSITY OF BRISTOL (UNIVBRIS)

AALTO-KORKEAKOULUSAATIO (AALTO)

UNIVERSITY OF PIRAEUS RESEARCH CENTER (UPRC)

ITALTEL SPA (ITL)

BIBA - BREMER INSTITUT FUER PRODUKTION UND LOGISTIK GMBH (BIBA).

Table of Contents

1 Introduction

The purpose of this deliverable is to document the MATILDA slicing metamodels. The term metamodels refers to two metamodels; namely the **Slice Intent Metamodel** and the **Slice Metamodel**. The Slice intent metamodel aims to represent all requirements that should be satisfied by a telco provider during the creation of a slice that will facilitate the deployment of an application. It should be noted that the application per se is represented by an analogous formal metamodel which is documented by Deliverable D1.2 [2] which is addressed as **Application Graph Metamodel**. The knowledge of this metamodel is considered a prerequisite for the comprehension of this deliverable.
An Application Graph consists of multiple components. These components may be hosted in various infrastructure depending on several requirements that are imposed by the "vertical" service provider. For example, a streaming service provider may require the placement of the applications transcoding components in a datacenter where horizontal scalability can be easily supported. Having done this any adaptation that may be required during runtime such the scale-out in a burst scenario can be easily supported. Furthermore, assuming that the profile of the vertical application is already known the amount of resources that have to reserved/allocated is predictable.
On the other hand, some components of this service may be required to be hosted on the edge. A content delivery component is an indicative example. Since the delay between the delivery node and the User Equipment (hereinafter UE) is a critical metric of a qualitative streaming service, its minimization can be achieved using specific placement options. Finally, the two aforementioned components may introduce additional restrictions such as minimum guaranteed throughput etc.
As it is easily inferred, the 5G provider has to facilitate the deployment of vertical applications that are highly distributed which are accompanied by multiple restrictions. These restrictions (also addressed as constraints within the document) must be expressed in a way which is unambiguous between the vertical service provider and the 5G-enabled telco provider. Resolving any unambiguity that may be raised implies the introduction of common representation model which will act as a shared conceptualization between the 'vertical' service provider and the telco. Hence, **the common representation model which will be used to facilitate the request of a vertical service provider to the telco is provided by the Slice Intent Metamodel**. Furthermore, **the common representation model which will be used to facilitate the response of a 5G-enabled telco provider to the vertical service provider is provided by the Slice Metamodel**.
It should be clarified that a specific serialization format has been selected in order to capture the semantics of the metamodels. This format is XSD[1]. However, any MATILDA adopter can transform this format to any domain specific language that is favorable. Furthermore, it should be also clarified that both modelling 'artefacts' are addressed as metamodels since they provide the means of creating infinite models (which are also addressed as model instances). Having said that, **any Slice Intent instance and any Slice instance must adhere to the respective metamodels**.

[1] https://www.w3.org/XML/Schema

The deliverable will shed light to both metamodels. Chapter 2 will summarize the usage of the metamodels within the MATILDA ecosystem by describing the flow from the point of view of the vertical service provider. Chapter 3 and chapter 4 elaborate and the Slice Intent and the Slice Metamodel respectively. Chapter 5 concludes this deliverable. It should be noted that Appendix 1 provides the formal XSD documentation.

2 The role of "Slice Intent" & Slice Metamodels in the MATILDA architecture

Before delving into the details of the Slice Intent and the Slice metamodels we will provide an overview of the usage of these models. The usage is graphically depicted on Figure 1 where the basic architectural components of the MATILDA framework and their relationship with the various models is provided.

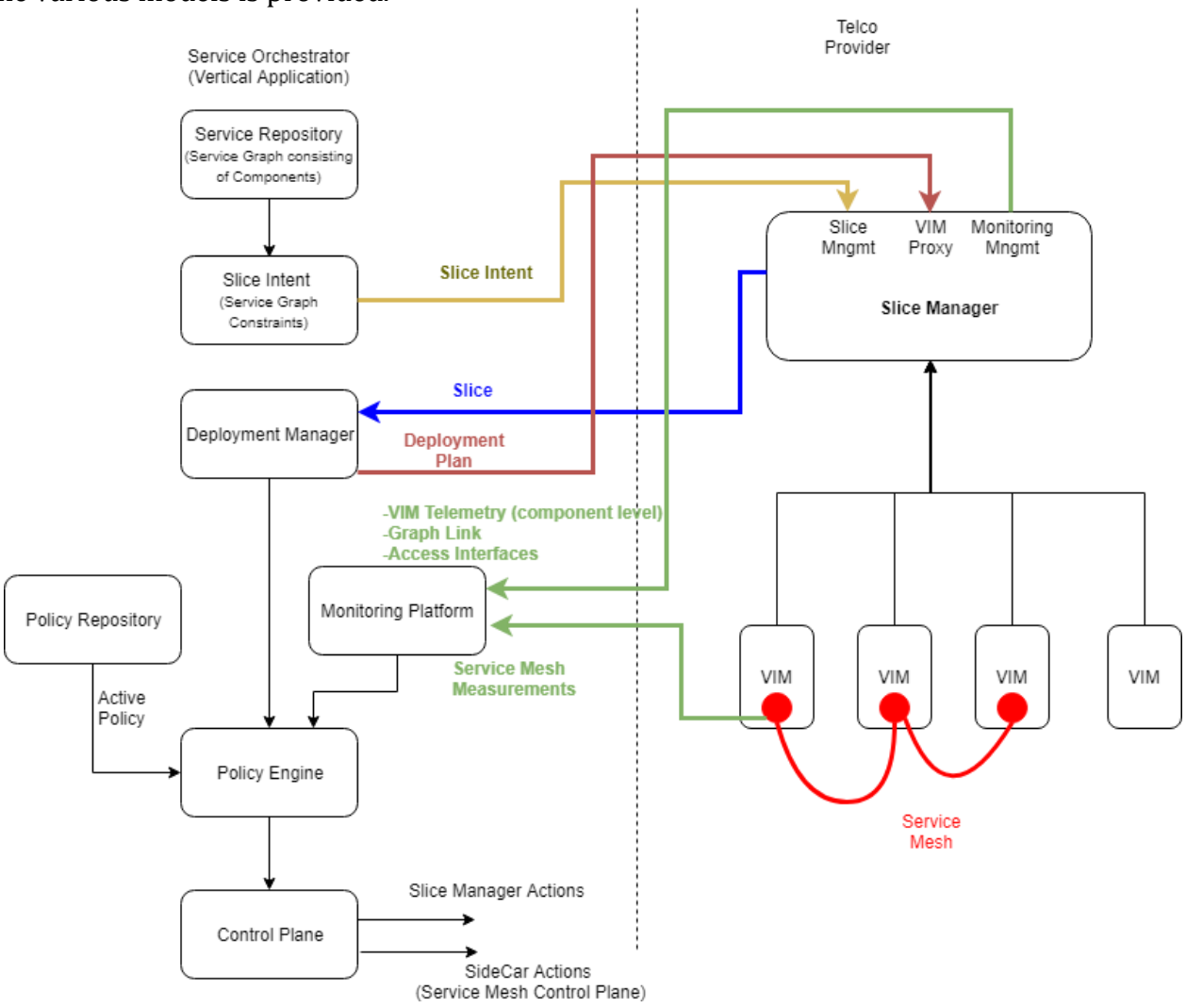


Figure 1 – Usage of MATILDA Metamodels

More specifically, as it is depicted, the flow starts with the selection of one Vertical Application which has to be deployed and supported by a telco provider. As clearly stated on the architectural deliverable (D1.1 [1]), MATILDA will support state of the art distributed applications. Therefore, a Vertical Application in MATILDA consists of multiple components that can be deployed on top of programmable infrastructure.
These components when combined to each other they formulate a **direct acyclic graph** (a.k.a. **DAG**) which represents a vertical application. In other words, a vertical application is represented by graph where components are vertexes and edges are the component-links. For the sake of clarity, MATILDA imposed a **formal metamodel** of this graph which is analyzed in Deliverable 1.2 [2]. Without delving into the details of this metamodel, it should be clarified that the components that comprise the graph should satisfy the requirements of **cloud-native applications**[2]. Being cloud-native is essential for the proper placement of these components to virtualized environments. Moreover, to be in-line with the state of the art principles of cloud applications, components should be also **programmable**. Programmability is provided by a **transparent proxy** which is running on top of the component. This proxy is also addressed as '**sidecar'**. A service graph that consists of programmable components is addressed as **Service Mesh**.
The **Service Repository** (see Figure 1) contains **all instances** of the service graphs that have been registered. The flow initiates by the selection of service graph by a service provider. As it is depicted on Figure 1, there are **two distinct administrative zones**. On the **left part** resides the administrative zone of the **service orchestrator** while on the right part the administrative zone of the **telco provider**. Hence, each administrative zone contains its own orchestration entity with clear responsibilities. The orchestration entity on the left is responsible to **instantiate a vertical application that meets specific requirements** on **the virtualized resources which will be prepared by the orchestration mechanism of the right**.
Taking under consideration the scope of the two orchestrators we can easily infer that the Service Orchestrator and the Telco Orchestration mechanisms follow a request/response pattern according to which the Service Orchestrator asks for a specific "setup" that is capable to satisfy some characteristics and the telco provider responds with the details of the environment that has to be used for the appropriate setup. The first request will be hereinafter addressed as **Slice Intent** while the latter as offered **Slice**. From the above definition, it is easily inferred that one service graph may lead to multiple Slice Intents since each instantiation may entail different requirements. Furthermore, the same Slice Intent may lead to multiple Slice instantiation by the telco provider according to the actual state of the provider.
The second step, after the selection of a service graph, relates to definition of the requirements that must be met during the deployment and the operation of the various components. These requirements are expressed in the form of constraints. Constraints may refer to the components of the service graph and to the graph links per se. This metamodel will be analyzed on "Chapter 3 - Overview of Slice Intent Metamode". The XSD notation will be used for the formal description of the metamodel; yet the metamodel may be converted to any notation is favorable by an adopter.

[2] https://12factor.net/

As a third step, the telco provider receives the slice intent and tries to find/create a proper setup which will satisfy the constraints that have been formulated in step 2. From a theoretical point of view, identifying a proper setup is equivalent to the solution of an optimization problem. However, most of the times constraint satisfaction problems are computational intractable. As a result, the response to a Slice Intent cannot be synchronous. Moreover, the cost (computational and time cost) for inferencing the most appropriate Slice may not desirable. Therefore, heuristic techniques will be employed that will target the near optimal identification of a solution in a reasonable (and meaningful) amount of time.
The solution that satisfies the constraints will be announced back to the Service Graph Orchestrator. The solution will be an instance of the Slice Metamodel which will be described on "Chapter 4 - Overview of Slice Metamodel". At this point, it should be clarified that a constraint of a Slice Intent may be hard constraint or soft constraint i.e. the solver that will find the optimal Slice "must" or "may" take these restrictions under consideration. After the proposal of one Slice by the telco provider, the fourth step includes the deployment and the configuration of the individual components to the infrastructural resources that the telco provider has allocated. As we will examine in the Chapter 4 such actions include the actual instantiation of the virtual machines and the proper configuration of the network interfaces. Upon instantiation of the Service Graph the vertical application is operational.
During the operational state of the service graph a set of independent, monitoring streams are aggregated by a monitoring server. As it is depicted, these streams relate a) to measurements that are provided by the telco provider per se (e.g. instrumentation of VIMs) and b) to measurements that are pulled by the Service Graph orchestrator. These measurements are directed to a Complex Event Processing (a.k.a. CEP) engine which is responsible to execute time-window-based operations in the form of rules. **The set of rules that are active per stream-evaluation is addressed as Policy**. Although many policies may be defined for one graph only one can be active.
Based on the rules' execution the orchestrator may fire some actions. These actions are classified in two categories. The first type of action can be realized by the telco provider itself. As depicted on Figure 1, the telco provider exposes a northbound interface to the service orchestrator which 'proxies' the functional capabilities of telco programmability. These capabilities span from allocation of virtualized resources (at the data center or at the edge level) to the provisioning of specific quality class on the network traffic between UEs and component interfaces. On the other hand, the second class of actions that will be supported are telco-agnostic. According to the Service Mesh paradigm each component that participates in the graph is programmable through a specific proxy. This proxy can handle multiple commands such as Layer-7 balancing etc. The type of conditions and actions that will be supported are describe on Deliverable 1.5 [4]. After examining the flow of the metamodels' usage we will delve into the details of the Slice Intent and the Slice metamodel respectively.

3 Overview of Slice Intent Metamodel

As depicted on Figure 2 a Slice Ident consists of four upper-level elements; namely a) an identifier that uniquely characterizes a request; b) a Service Mesh Identifier that uniquely characterizes a service graph; c) a set of constraints that have to be satisfied and d) a set of logical functions that have to be supported.

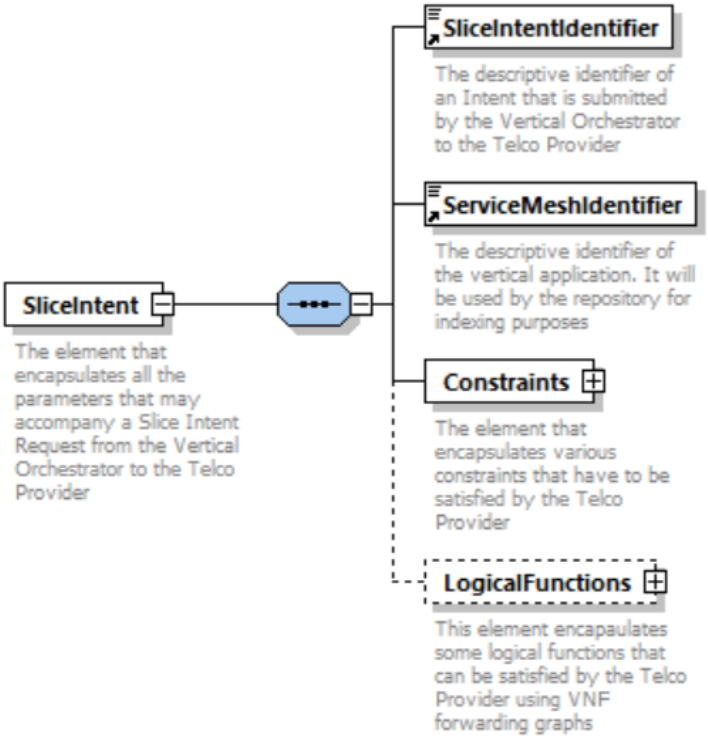


Figure 2 – Overview of the Slice Intent

While the two identifiers are self-explanatory the set of constraints have to be furtherly analyzed. Figure 3 illustrates the three types of constraints that will be supported.

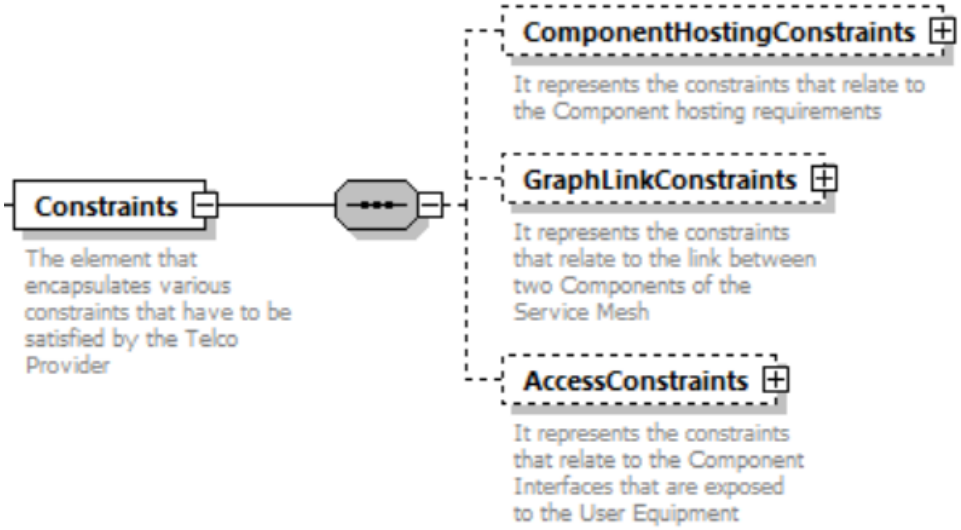


Figure 3 – High Level View of Constraints

The first set of constraints are the Component Hosting Constraints which are depicted on Figure 4.

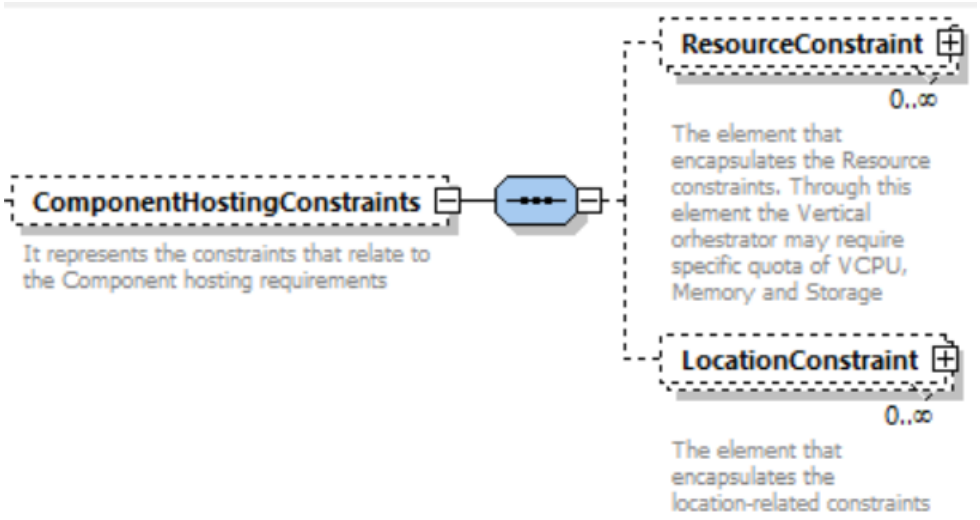


Figure 4 – The two types of Component Hosting Constraints

As depicted Component Hosting Constraints are furtherly grouped on Resource Constraints and Location Constraints. The detailed schema of the Resource Constraint is depicted on Figure 5.

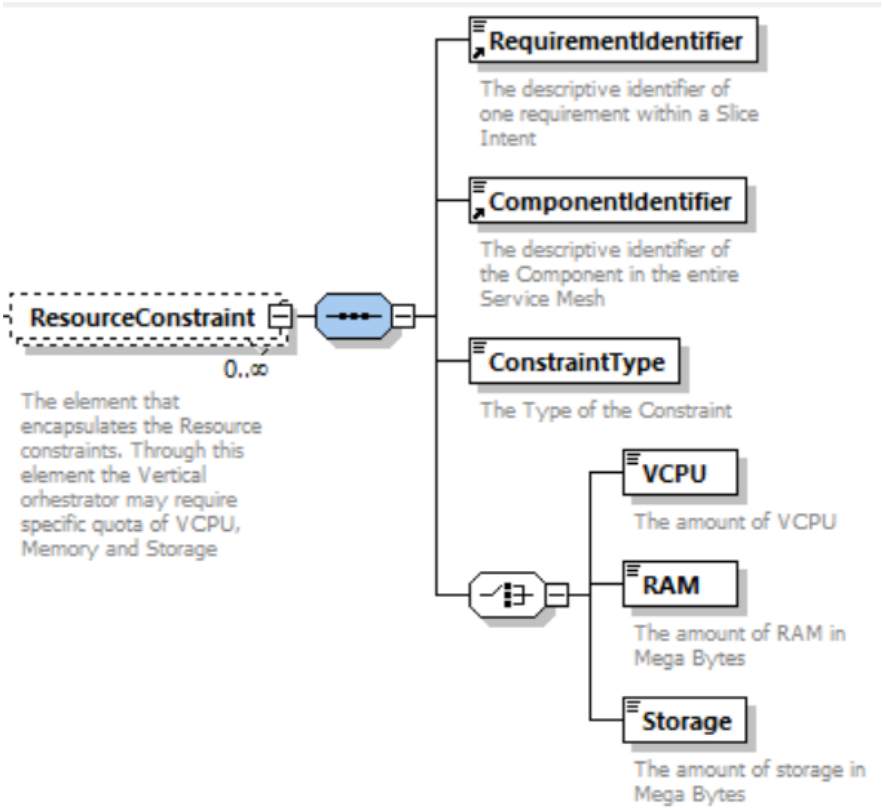


Figure 5 – The details of a Resource Constraint

As depicted, a resource constraint contains one descriptive identifier, which is used by the telco provider in order to report whether a constraint is fulfilled or not. In addition, a constraint has type which indicated whether a constraint is hard or soft. In fact, all constraints entail these basic attributes (i.e. identifier and type). A resource constraint is complemented by the amount of VCPUs, RAM and storage that is required per component.
On the other hand, the schema of the Location Constraint is depicted on Figure 6. As depicted, a location attribute can be declared per component.

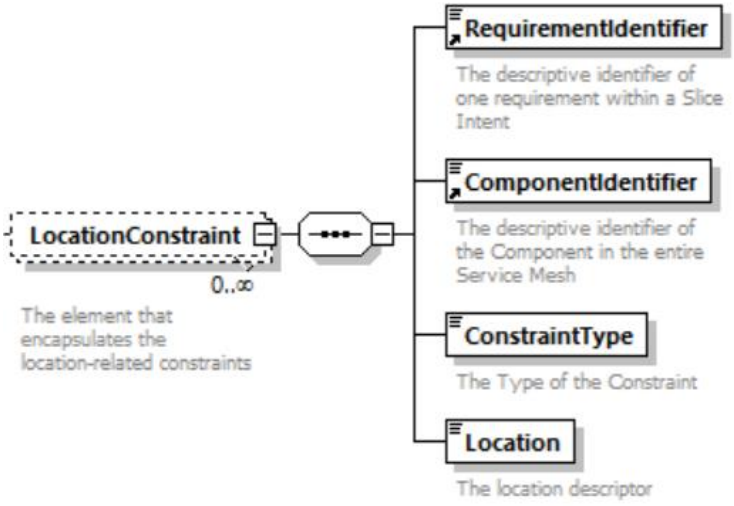


Figure 6 – The details of a Location Constraint

As already explained, Location and Resource constraints cover the Component Hosting constraints which is one of the constraint groups. The second group is the Graph Link constraints. Such constraints correspond to the requirements that have to be satisfied regarding the quality of the virtual link which is established between two components. During the deployment of a component all interfaces that he exposes are bound to network identifiers that are indicated by the provider.
The telco provider has the responsibility to perform the appropriate network programmability i.e. instantiating the appropriate VNF graphs and/or configuring the existing SDN switches. Through this configuration, the materialized link between components must satisfy some network requirements in terms of delay, jitter, packet loss and throughput. At this point, it should be clarified that a component that participates in a service mesh may expose two types of interfaces. The first type is the CORE and the other is the ACCESS type (see the Service Mesh metamodel of D1.2 [2]). CORE interfaces are the ones that are used among the components while ACCESS interfaces are the ones that interact with the UEs. It should be clear that Graph Link constraints refer to the interconnection of CORE interfaces only. Hence, the metamodel that is depicted in Figure 7 refers to CORE interfaces.
On the other hand, ACCESS interfaces entail a completely different metamodel. The metamodel is depicted on Figure 8 where the access constraints are visualized. More specifically, the type of constraints that can be provided relate to the QoS Class Identifier (a.k.a. QCI). QCI is a mechanism used in 3GPP Long Term Evolution (LTE) networks to ensure bearer traffic is allocated appropriate Quality of Service (QoS).

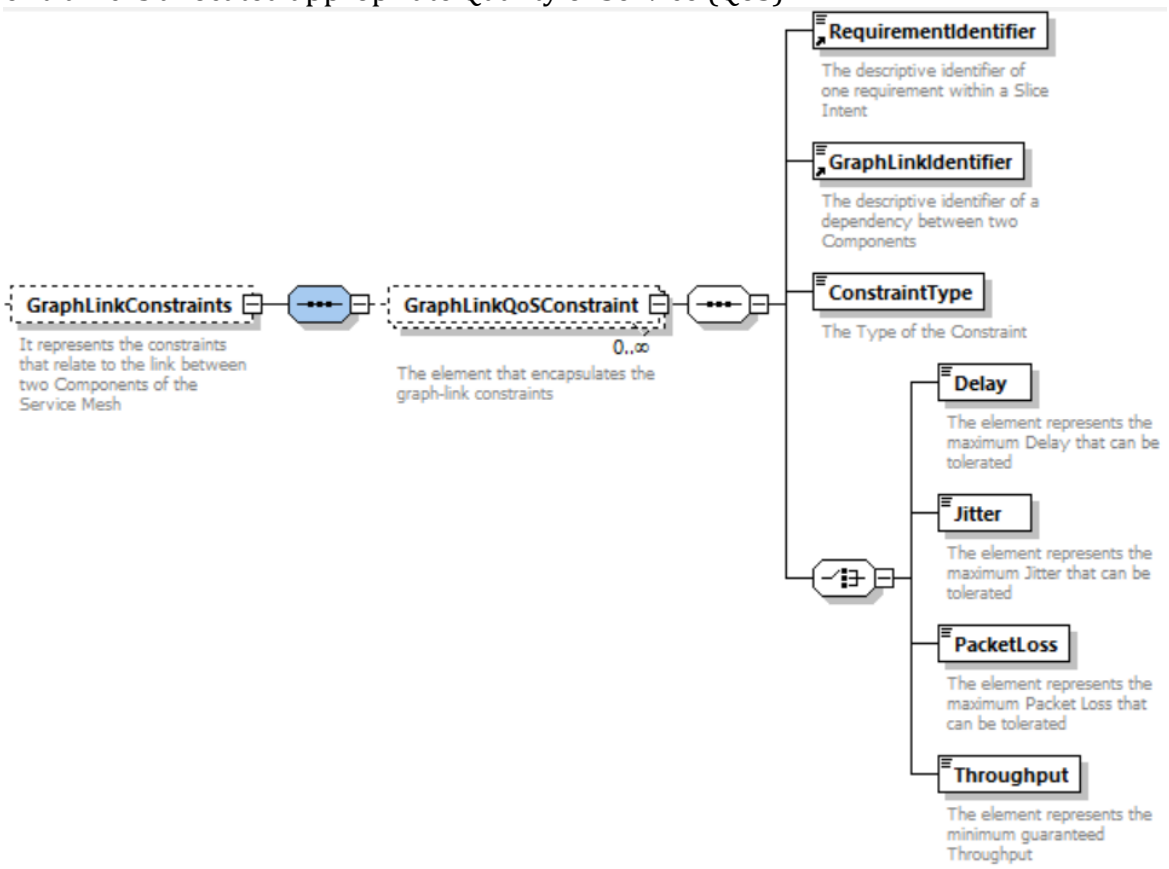


Figure 7 – The details of the Graph Link Constraints

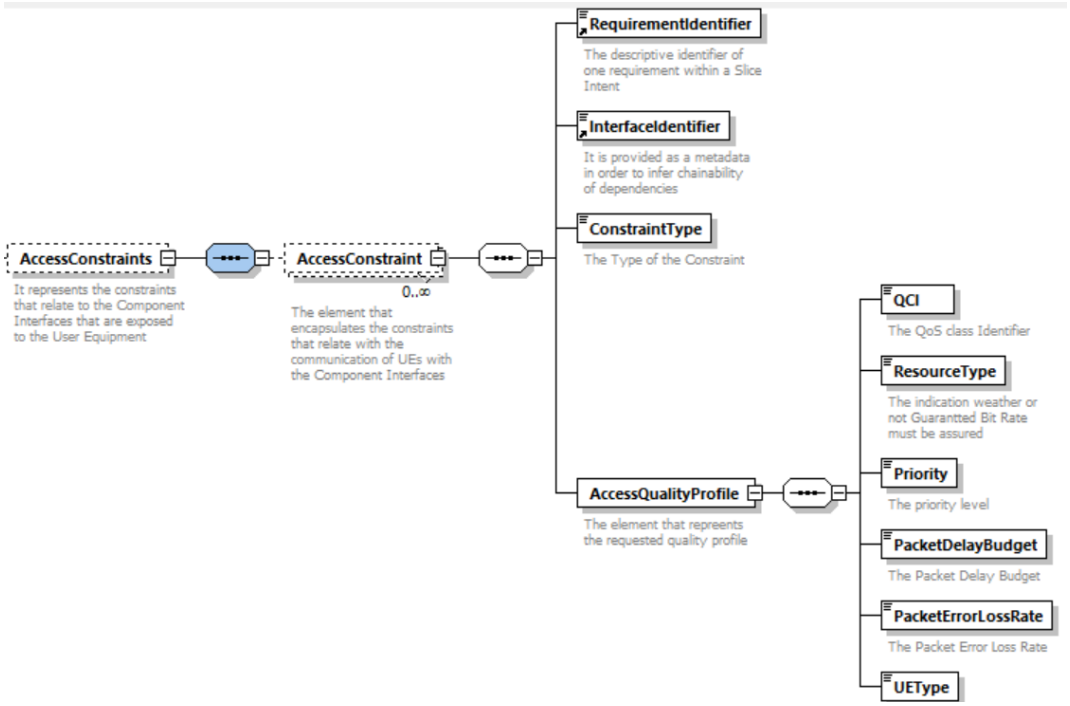


Figure 8 – The details of the Access Constraints

Finally, the last set of constraints that can be defined relate to specific logical functions that can be supported by a telco provider. In contradiction with the previous set of requirements, logical functions are not normative in the sense that a telco provider should perform an internal "translation" of this functions to internal mechanisms that can support it. For example, one telco provider may support an indicative firewalling requirement of one ACCESS interface using a properly configured VNF. To this end, the list of logical functions that are depicted on Figure 9 is purely indicative.

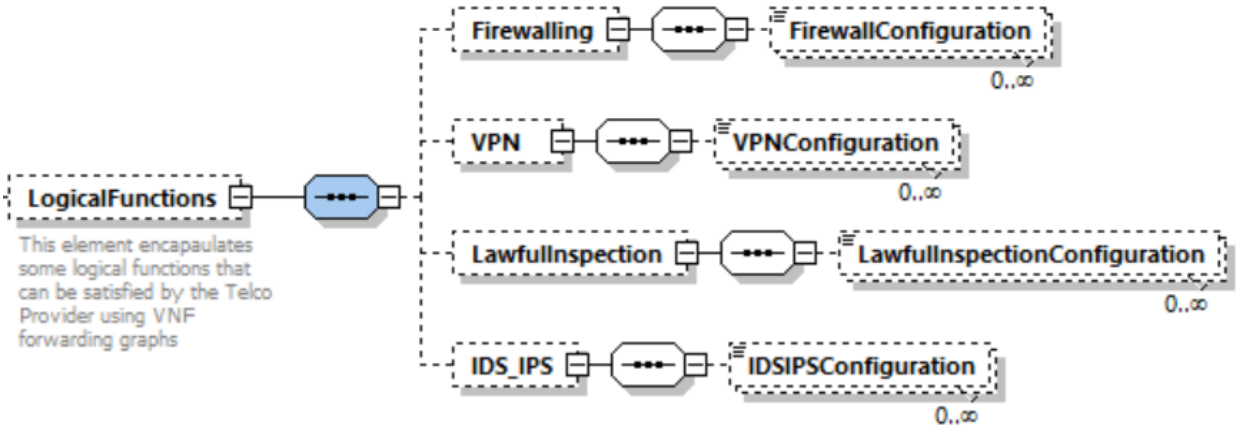


Figure 9 – Indicative Logical Functions

It should be noted that Annex I contains the reference documentation of the normative XSD schema elements.

4 Overview of Slice Metamodel

In the frame of the previous chapter, the Slice Intent has been described. As explained, the Slice Intent is a request in a request/response pattern between the Service Mesh Orchestrator and the telco provider (its northbound API). The Slice Metamodel represents the response of the provider. The high-level view of the Slice metamodel is depicted on Figure 10.

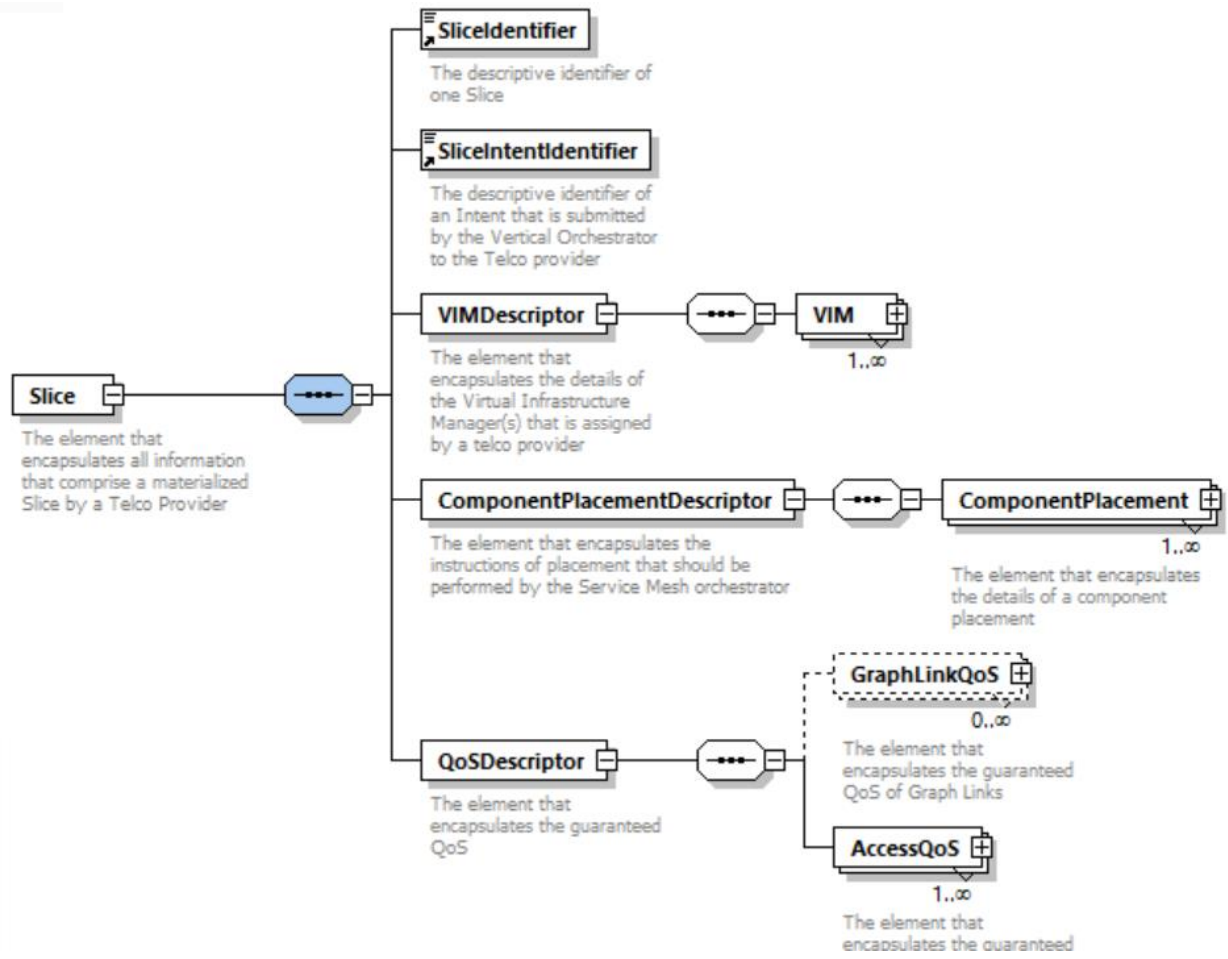


Figure 10 – Overview of Slice Metamodel

As depicted, the Slice consists of three main sections; namely the "**VIM Descriptor**", the "**Component Placement Descriptor**" and the "**QoS Descriptor**". The "VIM Descriptor" encapsulates the details of the Virtual Infrastructure Manager(s) that is/are assigned by a telco provider in order to satisfy the hosting requirements of a Service Mesh. The "Placement Descriptor" encapsulates the details of the placement plan that is indicated by the telco provider i.e. which component is hosted per VIM and which interface is bound to the assigned network identifiers. Finally, the "QoS Descriptor" encapsulates information regarding the guaranteed quality of access interfaces.

The overview of the VIM Descriptor is depicted on Figure 11.

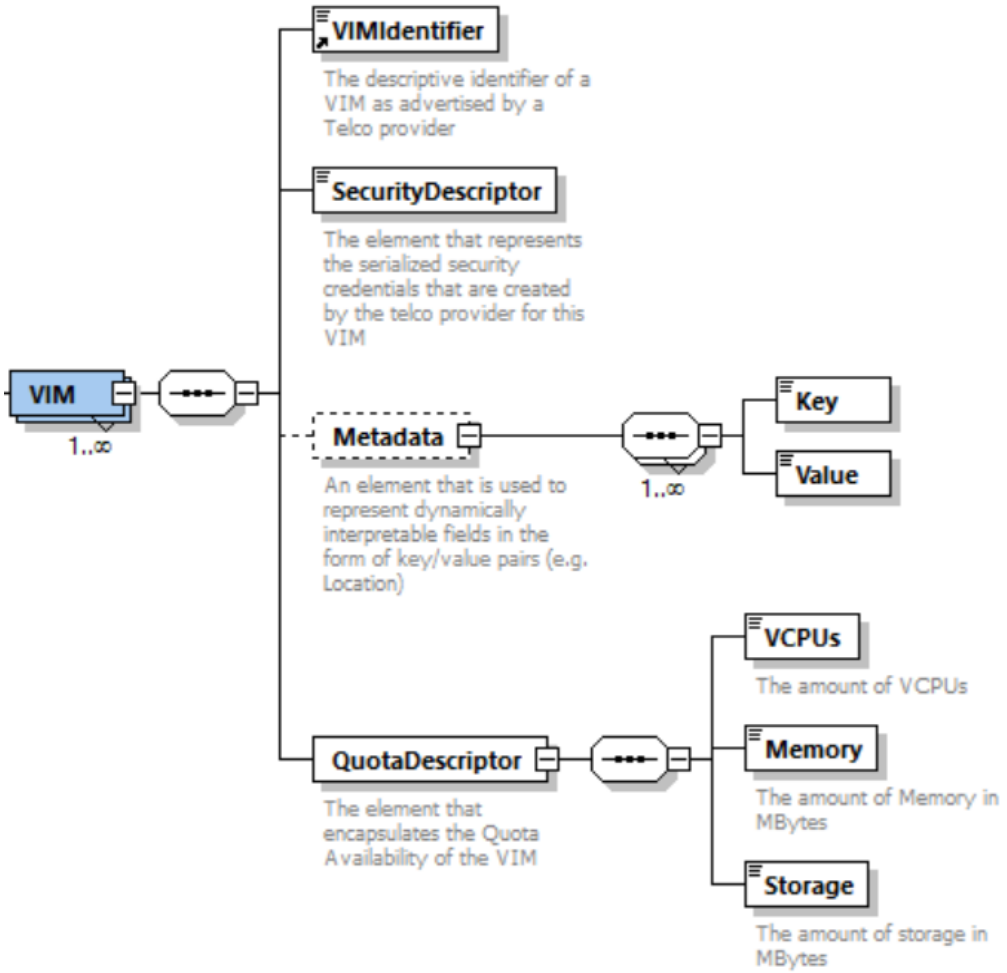


Figure 11 – Overview of VIM Descriptor

As depicted on Figure 11, a VIM Descriptor contains information regarding the authorization and the allocated quota that can be used by the Service Mesh orchestrator. The VIM Descriptor per se is not useful by the orchestrator if there is no knowledge about which component is installed in which VIM. This is addressed as placement information. Placement information is provided by the Placement Descriptor as depicted on Figure 12. Special emphasis should be given on the Interface Binding element. As already mentioned, a component has either an ACCESS interface or a CORE interface. Both interfaces when deployed they should bind to specific network-identifier that **are already prepared by the telco provider** prior to the creation of the Slice.

A link between two components can be characterized by a specific QoS level. This information is encapsulated by the "Graph Link Qos" element as depicted on Figure 13. However, these QoS parameters are not applicable for ACCESS interfaces. For these specific interfaces, the QoS characteristics that are defined on Figure 14 will be applied.

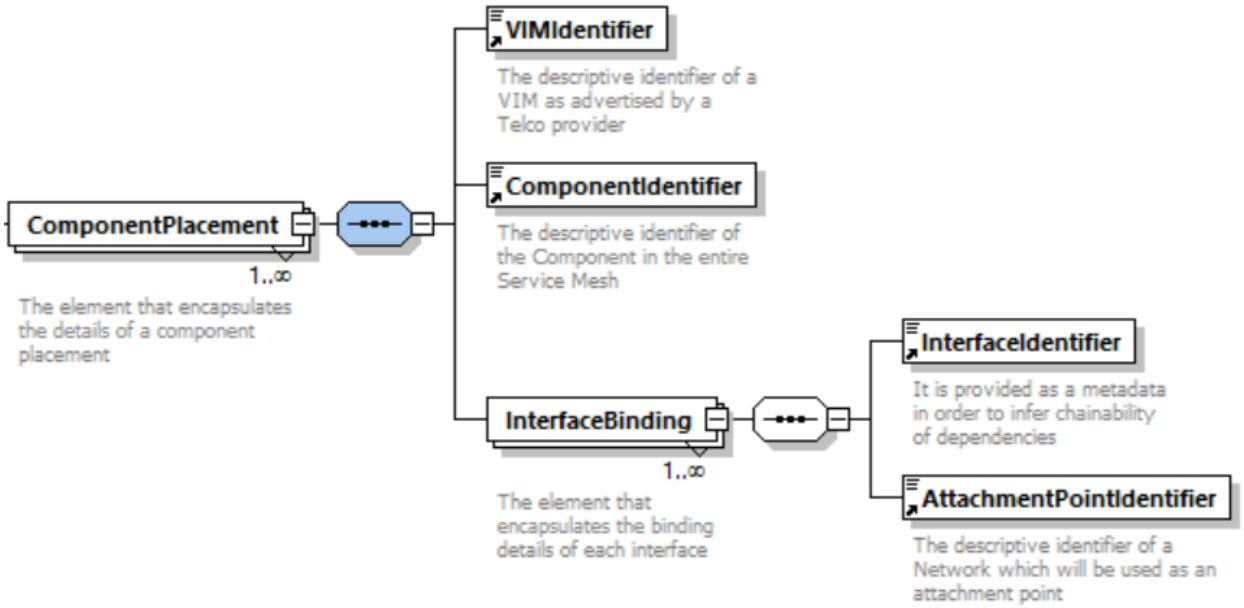


Figure 12 – Details of Component Placement

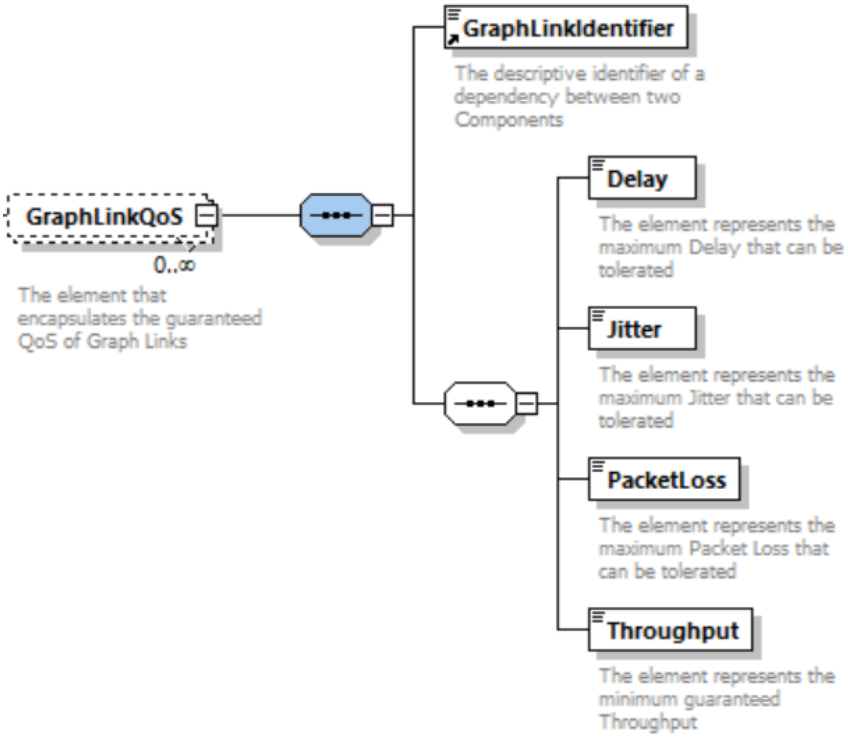


Figure 13 – Graph Link QoS characteristics

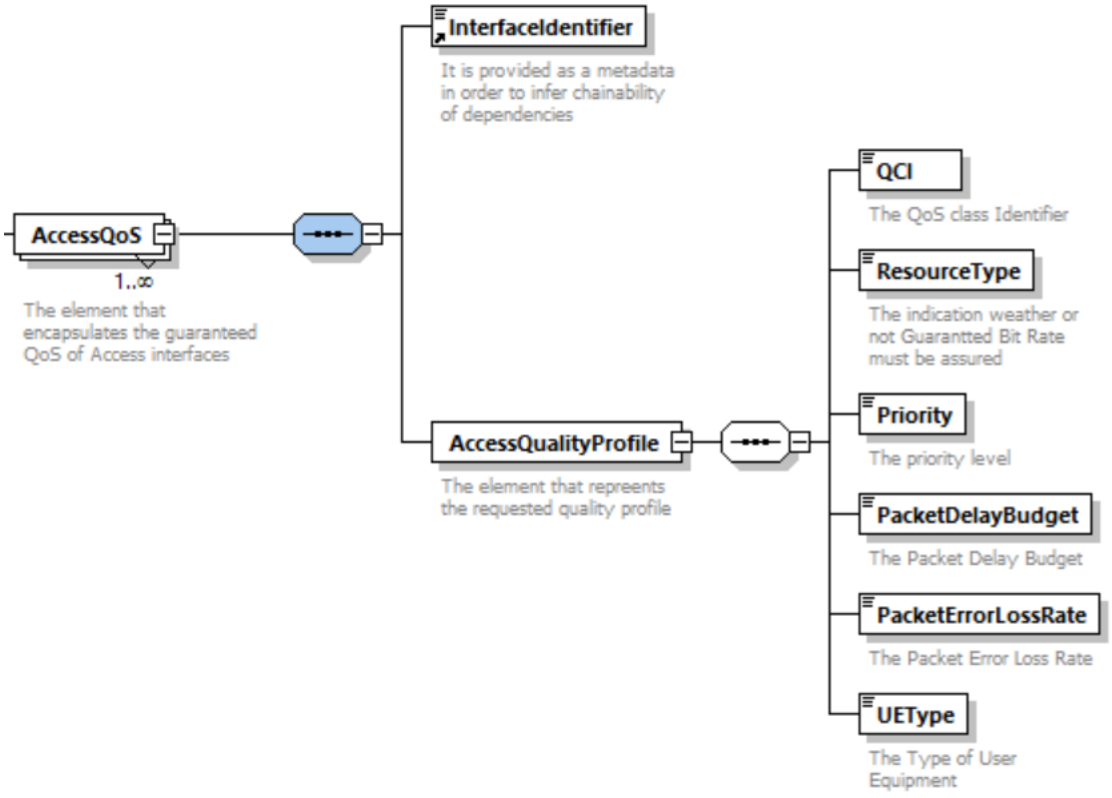


Figure 14 – Access QoS characteristics

Following the analogy of chapter 3, it should be noted that Annex I contains the reference documentation of the normative XSD schema elements.

5 Conclusions

The purpose of this deliverable is to elaborate on two metamodels of the MATILDA ecosystem which are the **Slice Intent** and the **Slice** metamodels. As already mentioned, both models act as a shared conceptualization between the vertical service provider and the telco provider. On the one hand, the Slice Intent is the common representation model which will be used to facilitate the request of a vertical service provider to the telco in order to deploy an application which consists of multiple components. On the other hand, the Slice metamodel acts as a common representation format which will be used to facilitate the response of a 5G-enabled telco provider to the vertical service provider. This response encapsulates all the information of the materialized slice that has been created/allocated by the telco provider.
In a nutshell, the Slice Intent consists of a set of **constraints that have to be fulfilled by the telco provider**. These constraints are characterized as soft and hard constraints based on the expectation that the vertical service provider has against the telco provider. It is self-explanatory that a soft constraint may be fulfilled while a hard constraint must be fulfilled. The fulfillment will rely on **constraint satisfaction techniques** which will try to **solve the theoretical problem of optimization** that is formulated by the aggregation of all the provided constraints.
The Slice Intent constraints are separated in "**Component Hosting Constraints**", in "**Graph Link Constraints**" and in "**Access Constraints**". Component Hosting Constraints encapsulate all requirements regarding the infrastructural resources of the deployed components. Such requirements include CPU, Memory, Hypervisor characteristics, location of deployment etc. The Graph Link Constraints encapsulate all networking requirements that relate to the communication of the components of the vertical application. Such networking requirements include delay, throughput, jitter etc. Finally, the Access Constraints contain the quality requirements of the component interfaces which will interact with the UE. Such requirements refer to the QCI characteristics per UE type.
Finally, the Slice Metamodel contains all parameters that should be known to an orchestrator which is responsible for the deployment and configuration of the vertical application. Such parameters include mainly the metadata of the allocated VIMs and the network-identifiers where the component identifiers should bind. The orchestrator of the vertical application will use this information in order to instantiate the virtualized components in the proper hypervisor and perform specific attachment of the instance interface(s) to specific network identifiers. The telco provider's orchestrator is responsible to perform proper configuration on all systems within its administrative zone (e.g. OSS,NFVO) in order for network flows to be terminated in the attachment points.
It should be noted that during the operation of the vertical application a slice reconfiguration might be required. To this end, an amended Slice Intent will be submitted to the telco provider which will take under consideration the new constraints. Upon satisfaction of these constraints an updated Slice will be sent back to the orchestrator of the vertical application. Hence, the asynchronous request/response pattern will always be followed. Finally, it should be clarified that this pattern is asynchronous because of the computational complexity of the constraint satisfaction problem that has to be solved. In order to tackle this complexity, MATILDA will utilize heuristic techniques that will accelerate the identification of a near-to-optimal solution.

6 References

[1] MATILDA D1.1 - MATILDA Framework and Reference Architecture

[2] MATILDA D1.2 - Chainable Application Component & 5G-ready Application Graph Metamodel

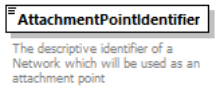
[3] MATILDA D1.3 - VNF/PNF & VNF Forwarding Graph Metamodel

[4] MATILDA D1.5 - Deployment and Runtime Policy Metamodel

Appendix 1: XSD Documentation

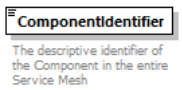
Elements

element **AttachmentPointIdentifier**

diagram	
used by	element **Slice/ComponentPlacementDescriptor/ComponentPlacement/InterfaceBinding**
annotation	documentation The descriptive identifier of a Network which will be used as an attachment point

source	`<xs:element name="AttachmentPointIdentifier">`
	`  <xs:annotation>`
	`  <xs:documentation>The descriptive identifier of a Network which will be used as an attachment point</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

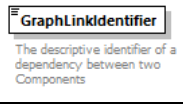
element **ComponentIdentifier**

diagram	
type	**xs:string**
properties	content simple
used by	elements **ServiceMesh/Component** **Slice/ComponentPlacementDescriptor/ComponentPlacement** **ServiceMesh/Component/RequiredInterface/GraphLink** **SliceIntent/Constraints/ComponentHostingConstraints/LocationConstraint** **SliceIntent/Constraints/ComponentHostingConstraints/ResourceConstraint**
annotation	documentation The descriptive identifier of the Component in the entire Service Mesh

source	`<xs:element name="ComponentIdentifier" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation>The descriptive identifier of the Component in the entire Service Mesh</xs:documentation>`

```
  </xs:annotation>
</xs:element>
```

element **GraphLinkIdentifier**

diagram	
type	**xs:string**
properties	content simple
used by	elements ServiceMesh/Component/RequiredInterface/GraphLink Slice/QoSDescriptor/GraphLinkQoS SliceIntent/Constraints/GraphLinkConstraints/GraphLinkQoSConstraint
annotation	documentation The descriptive identifier of a dependency between two Components

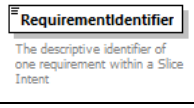
source	`<xs:element name="GraphLinkIdentifier" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation>The descriptive identifier of a dependency between two Components</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **InterfaceIdentifier**

diagram	
type	**xs:string**
properties	content simple
used by	elements SliceIntent/Constraints/AccessConstraints/AccessConstraint Slice/QoSDescriptor/AccessQoS ServiceMesh/Component/ExposedInterface ServiceMesh/Component/RequiredInterface/GraphLink Slice/ComponentPlacementDescriptor/ComponentPlacement/InterfaceBinding
annotation	documentation It is provided as a metadata in order to infer chainability of dependencies

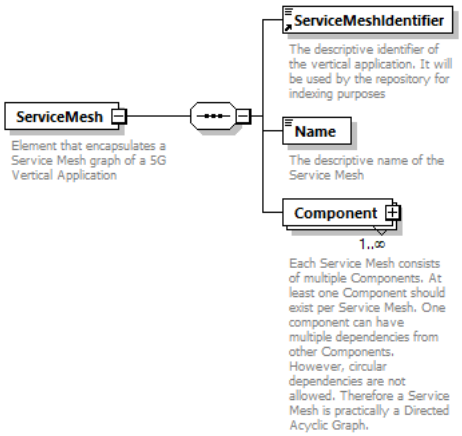
source	`<xs:element name="InterfaceIdentifier" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation>It is provided as a metadata in order to infer chainability of dependencies</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **RequirementIdentifier**

diagram	
type	**xs:string**
properties	content simple
used by	elements SliceIntent/Constraints/AccessConstraints/AccessConstraint

	SliceIntent/Constraints/GraphLinkConstraints/GraphLinkQoSConstraint SliceIntent/Constraints/ComponentHostingConstraints/LocationConstraint SliceIntent/Constraints/ComponentHostingConstraints/ResourceConstraint
annotation	documentation The descriptive identifier of one requirement within a Slice Intent
source	<xs:element name="RequirementIdentifier" type="xs:string"> <xs:annotation> <xs:documentation>The descriptive identifier of one requirement within a Slice Intent</xs:documentation> </xs:annotation> </xs:element>

element **ServiceMesh**

diagram	
properties	content complex
children	ServiceMeshIdentifier Name Component
annotation	documentation Element that encapsulates a Service Mesh graph of a 5G Vertical Application
source	<xs:element name="ServiceMesh"> <xs:annotation> <xs:documentation>Element that encapsulates a Service Mesh graph of a 5G Vertical Application</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="ServiceMeshIdentifier"> <xs:annotation> <xs:documentation>The descriptive identifier of the vertical application. It will be used by the repository for indexing purposes</xs:documentation> </xs:annotation> </xs:element> <xs:element name="Name" type="xs:string"> <xs:annotation> <xs:documentation>The descriptive name of the Service Mesh</xs:documentation> </xs:annotation> </xs:element> <xs:element name="Component" maxOccurs="unbounded">

```
        <xs:annotation>
          <xs:documentation>Each Service Mesh consists of multiple Components.
At least one Component should exist per Service Mesh. One component can have
multiple dependencies from other Components. However, circular dependencies
are not allowed. Therefore a Service Mesh is practically a Directed Acyclic
Graph.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="ComponentIdentifier">
              <xs:annotation>
                <xs:documentation>The descriptive identifier of the Component
in the entire Service Mesh</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="Distribution">
              <xs:annotation>
                <xs:documentation>The element that encapsulates the
information that is required for fetching an instance of a
Component</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="ImageDescriptor" type="xs:string">
                    <xs:annotation>
                      <xs:documentation>The descriptive identifier of the
Image/Container</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="RepositoryDescriptor" type="xs:string">
                    <xs:annotation>
                      <xs:documentation>The URI of the repository where the
component is located</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="ExposedInterface">
              <xs:annotation>
                <xs:documentation>It can refer to a single port or a range of
ports that serves this interface. One interface can be Access or Core
Type</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence maxOccurs="unbounded">
                  <xs:element ref="InterfaceIdentifier">
                    <xs:annotation>
                      <xs:documentation>The descriptive identifier of the
interface. It is required in order to infer chainability of dependencies
during the Service Mesh deployment.</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="InterfaceType">
                    <xs:annotation>
                      <xs:documentation>The classification of the exposed
```

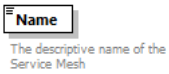
```
interface based on its positioning in the 5G network. It can be ACCESS or
CORE</xs:documentation>
                    </xs:annotation>
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:enumeration value="CORE"/>
                        <xs:enumeration value="ACCESS"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element name="Port">
                    <xs:annotation>
                      <xs:documentation>Refers to an internal port declaration
or range e.g. 4000 or 4000-4005</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="TransmissionProtocol" minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>Can be TCP or UDP</xs:documentation>
                    </xs:annotation>
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:enumeration value="TCP"/>
                        <xs:enumeration value="UDP"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="Configuration" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Set of environmental variables that should
be provided to the component during instantiation</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence maxOccurs="unbounded">
                  <xs:element name="Key" type="xs:string"/>
                  <xs:element name="Value" type="xs:string"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="Volume" minOccurs="0">
              <xs:annotation>
                <xs:documentation>The element that encapsulates the volume
mounting that has to be performed during the instantiation in the
hypervisor</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence minOccurs="0" maxOccurs="unbounded">
                  <xs:element name="VolumeType" type="xs:string"/>
                  <xs:element name="VolumeSource" type="xs:string"/>
                  <xs:element name="VolumeTarget" type="xs:string"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
```

```
            <xs:element name="MinimumExecutionRequirements" minOccurs="0">
              <xs:annotation>
                <xs:documentation>The element that encapsulates the minimum
requirements that have to be met by the hosting environment</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="VCPUs" type="xs:int" minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>Minimum amount of VCPUs that should be
provided by the hypervisor</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="RAM" minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>Minimum amount of RAM in
MBytes</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="Storage" minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>Minimum amount of Storage in
MBytes</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="HypervisorType" minOccurs="0">
                    <xs:annotation>
                      <xs:documentation>The type of Hypervisor that is
prefered</xs:documentation>
                    </xs:annotation>
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:enumeration value="ESXI"/>
                        <xs:enumeration value="KVM"/>
                        <xs:enumeration value="XEN"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="ExposedMetric" minOccurs="0">
              <xs:annotation>
                <xs:documentation>The list of Metrics that are reported by the
Component Sidecar</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence minOccurs="0" maxOccurs="unbounded">
                  <xs:element name="MetricIdentifier">
                    <xs:annotation>
                      <xs:documentation>The descriptiove identifier of the
Metric</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="MeasurementUnit">
                    <xs:annotation>
```

```
                        <xs:documentation>The Unit of
Measurement</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="RequiredInterface" minOccurs="0">
                <xs:annotation>
                  <xs:documentation>The element that encapsulates the required
dependencies of other components</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="GraphLink" maxOccurs="unbounded">
                      <xs:annotation>
                        <xs:documentation>Each dependency is modelled as a
GraphLink </xs:documentation>
                      </xs:annotation>
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element ref="GraphLinkIdentifier">
                            <xs:annotation>
                              <xs:documentation>The descriptive identifier of a
dependency between two Components</xs:documentation>
                            </xs:annotation>
                          </xs:element>
                          <xs:element ref="ComponentIdentifier">
                            <xs:annotation>
                              <xs:documentation>The descriptive identifier of
the component in the Service Mesh that satisfies the requirement. The
requestor is addressed as source (FROM) and the component that offers the
required interface is addressed as target (TO). This is the identifier of the
target.</xs:documentation>
                            </xs:annotation>
                          </xs:element>
                          <xs:element ref="InterfaceIdentifier">
                            <xs:annotation>
                              <xs:documentation>The descriptive identifier of
the target Component interface that is required</xs:documentation>
                            </xs:annotation>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="Capability" minOccurs="0">
                <xs:annotation>
                  <xs:documentation>The element that encapsulates Runtime
capabilities of the Components that are considered inherent</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Scaling" minOccurs="0">
```

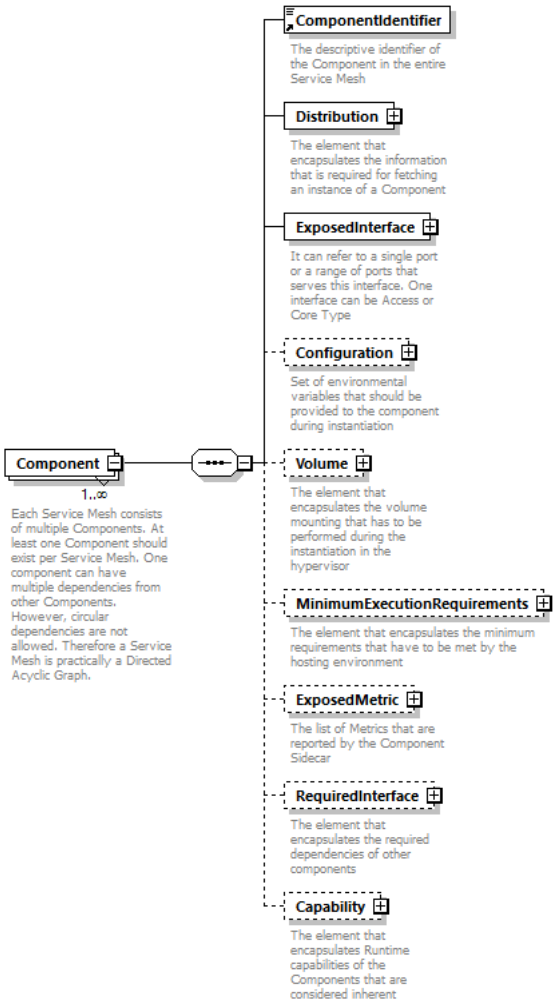
```
                    <xs:annotation>
                      <xs:documentation>Scaling can be Horizontal (if the
Component is completely stateless), Vertical if the Component is statefull or
Diagonal (i.e. both) in case of stateless Components</xs:documentation>
                    </xs:annotation>
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:enumeration value="HORIZONTAL"/>
                        <xs:enumeration value="VERTICAL"/>
                        <xs:enumeration value="DIAGONAL"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **ServiceMesh/Name**

diagram	
type	**xs:string**
properties	content simple
annotation	documentation The descriptive name of the Service Mesh

source	`<xs:element name="Name" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation>The descriptive name of the Service Mesh</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **ServiceMesh/Component**

diagram	
properties	minOcc 1 maxOcc unbounded content complex
children	**ComponentIdentifier Distribution ExposedInterface Configuration Volume MinimumExecutionRequirements ExposedMetric RequiredInterface Capability**
annotation	documentation Each Service Mesh consists of multiple Components. At least one Component should exist per Service Mesh. One component can have multiple dependencies from other Components. However, circular dependencies are not allowed. Therefore a Service Mesh is practically a Directed Acyclic Graph.
source	`<xs:element name="Component" maxOccurs="unbounded">` `  <xs:annotation>` ` <xs:documentation>Each Service Mesh consists of multiple Components. At least one Component should exist per Service Mesh. One component can have multiple dependencies from other Components. However, circular dependencies are not allowed. Therefore a Service Mesh is practically a Directed Acyclic Graph.</xs:documentation>` ` </xs:annotation>` ` <xs:complexType>` ` <xs:sequence>` ` <xs:element ref="ComponentIdentifier">` ` <xs:annotation>` ` <xs:documentation>The descriptive identifier of the Component in the`

```
entire Service Mesh</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Distribution">
        <xs:annotation>
          <xs:documentation>The element that encapsulates the information that
is required for fetching an instance of a Component</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ImageDescriptor" type="xs:string">
              <xs:annotation>
                <xs:documentation>The descriptive identifier of the
Image/Container</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="RepositoryDescriptor" type="xs:string">
              <xs:annotation>
                <xs:documentation>The URI of the repository where the
component is located</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ExposedInterface">
        <xs:annotation>
          <xs:documentation>It can refer to a single port or a range of ports
that serves this interface. One interface can be Access or Core
Type</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence maxOccurs="unbounded">
            <xs:element ref="InterfaceIdentifier">
              <xs:annotation>
                <xs:documentation>The descriptive identifier of the interface.
It is required in order to infer chainability of dependencies during the
Service Mesh deployment.</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="InterfaceType">
              <xs:annotation>
                <xs:documentation>The classification of the exposed interface
based on its positioning in the 5G network. It can be ACCESS or
CORE</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="CORE"/>
                  <xs:enumeration value="ACCESS"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="Port">
              <xs:annotation>
                <xs:documentation>Refers to an internal port declaration or
```

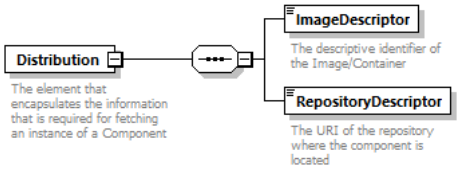
```
range e.g. 4000 or 4000-4005</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="TransmissionProtocol" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Can be TCP or UDP</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="TCP"/>
                  <xs:enumeration value="UDP"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Configuration" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Set of environmental variables that should be
provided to the component during instantiation</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence maxOccurs="unbounded">
            <xs:element name="Key" type="xs:string"/>
            <xs:element name="Value" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Volume" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The element that encapsulates the volume mounting
that has to be performed during the instantiation in the
hypervisor</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="VolumeType" type="xs:string"/>
            <xs:element name="VolumeSource" type="xs:string"/>
            <xs:element name="VolumeTarget" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="MinimumExecutionRequirements" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The element that encapsulates the minimum
requirements that have to be met by the hosting environment</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="VCPUs" type="xs:int" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Minimum amount of VCPUs that should be
provided by the hypervisor</xs:documentation>
              </xs:annotation>
            </xs:element>
```

```
            <xs:element name="RAM" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Minimum amount of RAM in
MBytes</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="Storage" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Minimum amount of Storage in
MBytes</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="HypervisorType" minOccurs="0">
              <xs:annotation>
                <xs:documentation>The type of Hypervisor that is
prefered</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="ESXI"/>
                  <xs:enumeration value="KVM"/>
                  <xs:enumeration value="XEN"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ExposedMetric" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The list of Metrics that are reported by the
Component Sidecar</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="MetricIdentifier">
              <xs:annotation>
                <xs:documentation>The descriptiove identifier of the
Metric</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="MeasurementUnit">
              <xs:annotation>
                <xs:documentation>The Unit of Measurement</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="RequiredInterface" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The element that encapsulates the required
dependencies of other components</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
```

```
            <xs:element name="GraphLink" maxOccurs="unbounded">
              <xs:annotation>
                <xs:documentation>Each dependency is modelled as a GraphLink
</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence>
                  <xs:element ref="GraphLinkIdentifier">
                    <xs:annotation>
                      <xs:documentation>The descriptive identifier of a
dependency between two Components</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element ref="ComponentIdentifier">
                    <xs:annotation>
                      <xs:documentation>The descriptive identifier of the
component in the Service Mesh that satisfies the requirement. The requestor is
addressed as source (FROM) and the component that offers the required
interface is addressed as target (TO). This is the identifier of the
target.</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element ref="InterfaceIdentifier">
                    <xs:annotation>
                      <xs:documentation>The descriptive identifier of the
target Component interface that is required</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Capability" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The element that encapsulates Runtime capabilities
of the Components that are considered inherent</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Scaling" minOccurs="0">
              <xs:annotation>
                <xs:documentation>Scaling can be Horizontal (if the Component
is completely stateless), Vertical if the Component is statefull or Diagonal
(i.e. both) in case of stateless Components</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="HORIZONTAL"/>
                  <xs:enumeration value="VERTICAL"/>
                  <xs:enumeration value="DIAGONAL"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
          </xs:sequence>
```

```
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **ServiceMesh/Component/Distribution**

diagram	
properties	content complex
children	ImageDescriptor RepositoryDescriptor
annotation	documentation The element that encapsulates the information that is required for fetching an instance of a Component
source	<xs:element name="Distribution"> <xs:annotation> <xs:documentation>The element that encapsulates the information that is required for fetching an instance of a Component</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element name="ImageDescriptor" type="xs:string"> <xs:annotation> <xs:documentation>The descriptive identifier of the Image/Container</xs:documentation> </xs:annotation> </xs:element> <xs:element name="RepositoryDescriptor" type="xs:string"> <xs:annotation> <xs:documentation>The URI of the repository where the component is located</xs:documentation> </xs:annotation> </xs:element> </xs:sequence> </xs:complexType> </xs:element>

element **ServiceMesh/Component/Distribution/ImageDescriptor**

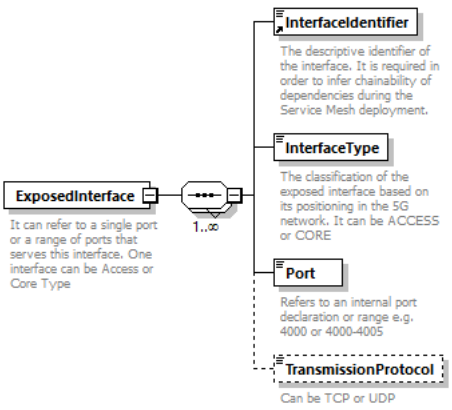
diagram	
type	**xs:string**
properties	content simple
annotation	documentation The descriptive identifier of the Image/Container
source	<xs:element name="ImageDescriptor" type="xs:string">

	<xs:annotation> <xs:documentation>The descriptive identifier of the Image/Container</xs:documentation> </xs:annotation> </xs:element>

element **ServiceMesh/Component/Distribution/RepositoryDescriptor**

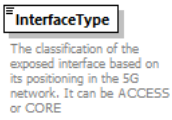
diagram	
type	**xs:string**
properties	content simple
annotation	documentation The URI of the repository where the component is located
source	<xs:element name="RepositoryDescriptor" type="xs:string"> <xs:annotation> <xs:documentation>The URI of the repository where the component is located</xs:documentation> </xs:annotation> </xs:element>

element **ServiceMesh/Component/ExposedInterface**

diagram	
properties	content complex
children	**InterfaceIdentifier InterfaceType Port TransmissionProtocol**
annotation	documentation It can refer to a single port or a range of ports that serves this interface. One interface can be Access or Core Type
source	<xs:element name="ExposedInterface"> <xs:annotation> <xs:documentation>It can refer to a single port or a range of ports that serves this interface. One interface can be Access or Core Type</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence maxOccurs="unbounded"> <xs:element ref="InterfaceIdentifier">

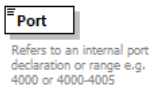
```
        <xs:annotation>
          <xs:documentation>The descriptive identifier of the interface. It is
required in order to infer chainability of dependencies during the Service
Mesh deployment.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="InterfaceType">
        <xs:annotation>
          <xs:documentation>The classification of the exposed interface based
on its positioning in the 5G network. It can be ACCESS or
CORE</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="CORE"/>
            <xs:enumeration value="ACCESS"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Port">
        <xs:annotation>
          <xs:documentation>Refers to an internal port declaration or range
e.g. 4000 or 4000-4005</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="TransmissionProtocol" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Can be TCP or UDP</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="TCP"/>
            <xs:enumeration value="UDP"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **ServiceMesh/Component/ExposedInterface/InterfaceType**

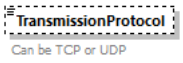
diagram	
type	restriction of **xs:string**
properties	content simple
facets	Kind Value Annotation enumeration CORE enumeration ACCESS
annotation	documentation The classification of the exposed interface based on its positioning in the 5G network. It can be ACCESS or CORE

source	<xs:element name="InterfaceType"> <xs:annotation> <xs:documentation>The classification of the exposed interface based on its positioning in the 5G network. It can be ACCESS or CORE</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="CORE"/> <xs:enumeration value="ACCESS"/> </xs:restriction> </xs:simpleType> </xs:element>

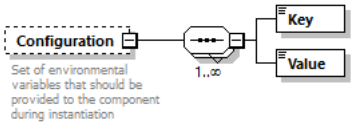
element **ServiceMesh/Component/ExposedInterface/Port**

diagram	
annotation	documentation Refers to an internal port declaration or range e.g. 4000 or 4000-4005
source	<xs:element name="Port"> <xs:annotation> <xs:documentation>Refers to an internal port declaration or range e.g. 4000 or 4000-4005</xs:documentation> </xs:annotation> </xs:element>

element **ServiceMesh/Component/ExposedInterface/TransmissionProtocol**

diagram	
type	restriction of **xs:string**
properties	minOcc 0 maxOcc 1 content simple
facets	Kind Value Annotation enumeration TCP enumeration UDP
annotation	documentation Can be TCP or UDP
source	<xs:element name="TransmissionProtocol" minOccurs="0"> <xs:annotation> <xs:documentation>Can be TCP or UDP</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="TCP"/> <xs:enumeration value="UDP"/> </xs:restriction> </xs:simpleType> </xs:element>

element **ServiceMesh/Component/Configuration**

diagram	
properties	minOcc 0 maxOcc 1 content complex
children	**Key Value**
annotation	documentation Set of environmental variables that should be provided to the component during instantiation

source	`<xs:element name="Configuration" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation>Set of environmental variables that should be provided to the component during instantiation</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`  <xs:sequence maxOccurs="unbounded">`
	`  <xs:element name="Key" type="xs:string"/>`
	`  <xs:element name="Value" type="xs:string"/>`
	`  </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

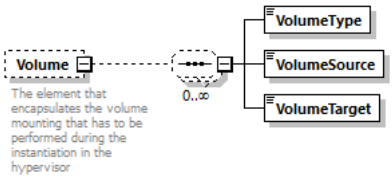
element **ServiceMesh/Component/Configuration/Key**

diagram	
type	**xs:string**
properties	content simple
source	`<xs:element name="Key" type="xs:string"/>`

element **ServiceMesh/Component/Configuration/Value**

diagram	
type	**xs:string**
properties	content simple
source	`<xs:element name="Value" type="xs:string"/>`

element **ServiceMesh/Component/Volume**

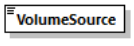
diagram	
properties	minOcc 0 maxOcc 1 content complex
children	**VolumeType VolumeSource VolumeTarget**
annotation	documentation The element that encapsulates the volume mounting that has to be performed during the instantiation in the hypervisor

source	`<xs:element name="Volume" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation>The element that encapsulates the volume mounting that has to be performed during the instantiation in the hypervisor</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`  <xs:sequence minOccurs="0" maxOccurs="unbounded">`
	`  <xs:element name="VolumeType" type="xs:string"/>`
	`  <xs:element name="VolumeSource" type="xs:string"/>`
	`  <xs:element name="VolumeTarget" type="xs:string"/>`
	`  </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

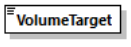
element **ServiceMesh/Component/Volume/VolumeType**

diagram	
type	**xs:string**
properties	content simple
source	`<xs:element name="VolumeType" type="xs:string"/>`

element **ServiceMesh/Component/Volume/VolumeSource**

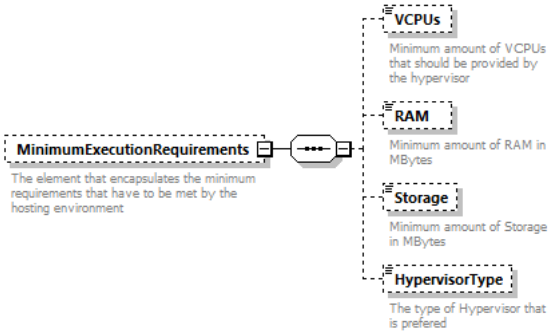
diagram	
type	**xs:string**
properties	content simple
source	`<xs:element name="VolumeSource" type="xs:string"/>`

element **ServiceMesh/Component/Volume/VolumeTarget**

diagram	

type	**xs:string**
properties	content simple
source	`<xs:element name="VolumeTarget" type="xs:string"/>`

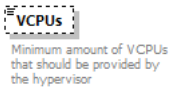
element **ServiceMesh/Component/MinimumExecutionRequirements**

diagram	
properties	minOcc 0 maxOcc 1 content complex
children	**VCPUs RAM Storage HypervisorType**
annotation	documentation The element that encapsulates the minimum requirements that have to be met by the hosting environment
source	see below

```
<xs:element name="MinimumExecutionRequirements" minOccurs="0">
  <xs:annotation>
    <xs:documentation>The element that encapsulates the minimum requirements
that have to be met by the hosting environment</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VCPUs" type="xs:int" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Minimum amount of VCPUs that should be provided by
the hypervisor</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="RAM" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Minimum amount of RAM in MBytes</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Storage" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Minimum amount of Storage in
MBytes</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="HypervisorType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>The type of Hypervisor that is
prefered</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
```

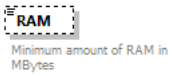
```
          <xs:restriction base="xs:string">
            <xs:enumeration value="ESXI"/>
            <xs:enumeration value="KVM"/>
            <xs:enumeration value="XEN"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **ServiceMesh/Component/MinimumExecutionRequirements/VCPUs**

diagram	
type	**xs:int**
properties	minOcc 0 maxOcc 1 content simple
annotation	documentation Minimum amount of VCPUs that should be provided by the hypervisor

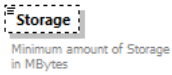
source	`<xs:element name="VCPUs" type="xs:int" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation>Minimum amount of VCPUs that should be provided by the hypervisor</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **ServiceMesh/Component/MinimumExecutionRequirements/RAM**

diagram	
properties	minOcc 0 maxOcc 1
annotation	documentation Minimum amount of RAM in MBytes

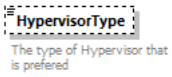
source	`<xs:element name="RAM" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation>Minimum amount of RAM in MBytes</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **ServiceMesh/Component/MinimumExecutionRequirements/Storage**

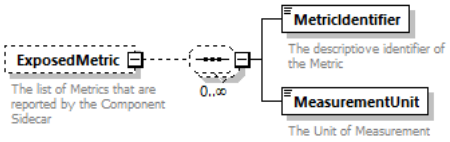
diagram	
properties	minOcc 0 maxOcc 1
annotation	documentation

	Minimum amount of Storage in MBytes
source	<xs:element name="Storage" minOccurs="0"> <xs:annotation> <xs:documentation>Minimum amount of Storage in MBytes</xs:documentation> </xs:annotation> </xs:element>

element **ServiceMesh/Component/MinimumExecutionRequirements/HypervisorType**

diagram	
type	restriction of **xs:string**
properties	minOcc 0 maxOcc 1 content simple
facets	Kind Value Annotation enumeration ESXI enumeration KVM enumeration XEN
annotation	documentation The type of Hypervisor that is prefered
source	<xs:element name="HypervisorType" minOccurs="0"> <xs:annotation> <xs:documentation>The type of Hypervisor that is prefered</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="ESXI"/> <xs:enumeration value="KVM"/> <xs:enumeration value="XEN"/> </xs:restriction> </xs:simpleType> </xs:element>

element **ServiceMesh/Component/ExposedMetric**

diagram	
properties	minOcc 0 maxOcc 1 content complex
children	**MetricIdentifier** **MeasurementUnit**
annotation	documentation The list of Metrics that are reported by the Component Sidecar
source	<xs:element name="ExposedMetric" minOccurs="0"> <xs:annotation> <xs:documentation>The list of Metrics that are reported by the Component

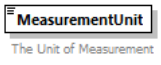
```
Sidecar</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="MetricIdentifier">
        <xs:annotation>
          <xs:documentation>The descriptiove identifier of the
Metric</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="MeasurementUnit">
        <xs:annotation>
          <xs:documentation>The Unit of Measurement</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **ServiceMesh/Component/ExposedMetric/MetricIdentifier**

diagram	
annotation	documentation The descriptiove identifier of the Metric

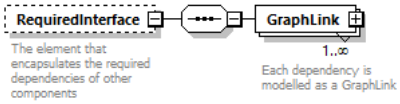
source	`<xs:element name="MetricIdentifier">`
	`  <xs:annotation>`
	`  <xs:documentation>The descriptiove identifier of the Metric</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **ServiceMesh/Component/ExposedMetric/MeasurementUnit**

diagram	
annotation	documentation The Unit of Measurement

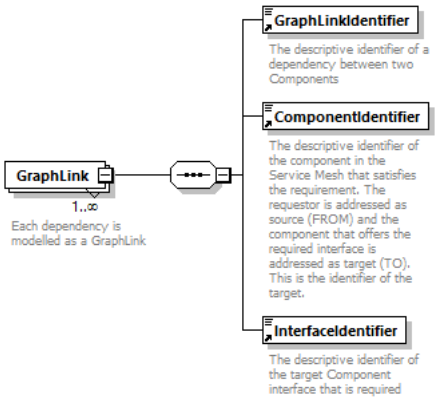
source	`<xs:element name="MeasurementUnit">`
	`  <xs:annotation>`
	`  <xs:documentation>The Unit of Measurement</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **ServiceMesh/Component/RequiredInterface**

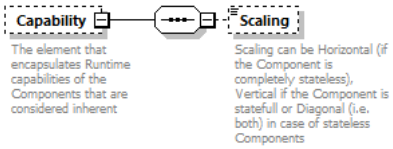
diagram	
properties	minOcc 0

	maxOcc 1 content complex
children	**GraphLink**
annotation	documentation The element that encapsulates the required dependencies of other components
source	<xs:element name="RequiredInterface" minOccurs="0"> <xs:annotation> <xs:documentation>The element that encapsulates the required dependencies of other components</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element name="GraphLink" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>Each dependency is modelled as a GraphLink</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="GraphLinkIdentifier"> <xs:annotation> <xs:documentation>The descriptive identifier of a dependency between two Components</xs:documentation> </xs:annotation> </xs:element> <xs:element ref="ComponentIdentifier"> <xs:annotation> <xs:documentation>The descriptive identifier of the component in the Service Mesh that satisfies the requirement. The requestor is addressed as source (FROM) and the component that offers the required interface is addressed as target (TO). This is the identifier of the target.</xs:documentation> </xs:annotation> </xs:element> <xs:element ref="InterfaceIdentifier"> <xs:annotation> <xs:documentation>The descriptive identifier of the target Component interface that is required</xs:documentation> </xs:annotation> </xs:element> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element>

element **ServiceMesh/Component/RequiredInterface/GraphLink**

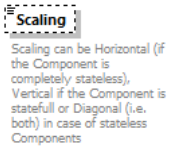
diagram	
properties	minOcc 1 maxOcc unbounded content complex
children	**GraphLinkIdentifier** **ComponentIdentifier** **InterfaceIdentifier**
annotation	documentation Each dependency is modelled as a GraphLink
source	<xs:element name="GraphLink" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>Each dependency is modelled as a GraphLink </xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="GraphLinkIdentifier"> <xs:annotation> <xs:documentation>The descriptive identifier of a dependency between two Components</xs:documentation> </xs:annotation> </xs:element> <xs:element ref="ComponentIdentifier"> <xs:annotation> <xs:documentation>The descriptive identifier of the component in the Service Mesh that satisfies the requirement. The requestor is addressed as source (FROM) and the component that offers the required interface is addressed as target (TO). This is the identifier of the target.</xs:documentation> </xs:annotation> </xs:element> <xs:element ref="InterfaceIdentifier"> <xs:annotation> <xs:documentation>The descriptive identifier of the target Component interface that is required</xs:documentation> </xs:annotation> </xs:element> </xs:sequence> </xs:complexType> </xs:element>

element **ServiceMesh/Component/Capability**

diagram	
properties	minOcc 0 maxOcc 1 content complex
children	**Scaling**
annotation	documentation The element that encapsulates Runtime capabilities of the Components that are considered inherent

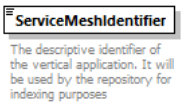
source	`<xs:element name="Capability" minOccurs="0">`
	`<xs:annotation>`
	`<xs:documentation>The element that encapsulates Runtime capabilities of the Components that are considered inherent</xs:documentation>`
	`</xs:annotation>`
	`<xs:complexType>`
	`<xs:sequence>`
	`<xs:element name="Scaling" minOccurs="0">`
	`<xs:annotation>`
	`<xs:documentation>Scaling can be Horizontal (if the Component is completely stateless), Vertical if the Component is statefull or Diagonal (i.e. both) in case of stateless Components</xs:documentation>`
	`</xs:annotation>`
	`<xs:simpleType>`
	`<xs:restriction base="xs:string">`
	`<xs:enumeration value="HORIZONTAL"/>`
	`<xs:enumeration value="VERTICAL"/>`
	`<xs:enumeration value="DIAGONAL"/>`
	`</xs:restriction>`
	`</xs:simpleType>`
	`</xs:element>`
	`</xs:sequence>`
	`</xs:complexType>`
	`</xs:element>`

element **ServiceMesh/Component/Capability/Scaling**

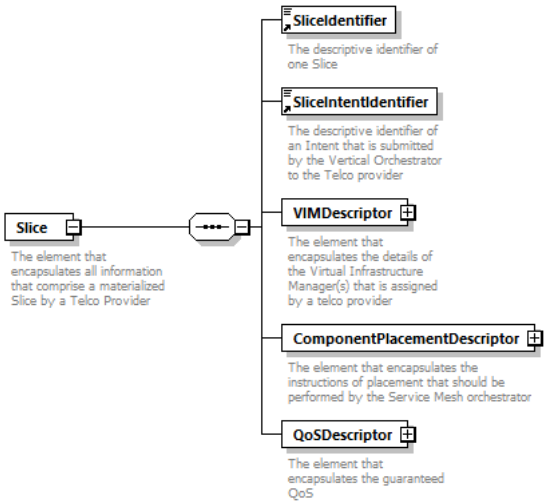
diagram	
type	restriction of **xs:string**
properties	minOcc 0 maxOcc 1 content simple
facets	Kind / Value / Annotation enumeration HORIZONTAL enumeration VERTICAL enumeration DIAGONAL
annotation	documentation

	Scaling can be Horizontal (if the Component is completely stateless), Vertical if the Component is statefull or Diagonal (i.e. both) in case of stateless Components
source	<xs:element name="Scaling" minOccurs="0"> <xs:annotation> <xs:documentation>Scaling can be Horizontal (if the Component is completely stateless), Vertical if the Component is statefull or Diagonal (i.e. both) in case of stateless Components</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="HORIZONTAL"/> <xs:enumeration value="VERTICAL"/> <xs:enumeration value="DIAGONAL"/> </xs:restriction> </xs:simpleType> </xs:element>

element **ServiceMeshIdentifier**

diagram	
type	**xs:string**
properties	content simple
used by	elements **ServiceMesh SliceIntent**
annotation	documentation The descriptive identifier of the vertical application. It will be used by the repository for indexing purposes
source	<xs:element name="ServiceMeshIdentifier" type="xs:string"> <xs:annotation> <xs:documentation>The descriptive identifier of the vertical application. It will be used by the repository for indexing purposes</xs:documentation> </xs:annotation> </xs:element>

element **Slice**

diagram	
properties	content complex
children	SliceIdentifier SliceIntentIdentifier VIMDescriptor ComponentPlacementDescriptor QoSDescriptor
annotation	documentation The element that encapsulates all information that comprise a materialized Slice by a Telco Provider
source	<xs:element name="Slice"> <xs:annotation> <xs:documentation>The element that encapsulates all information that comprise a materialized Slice by a Telco Provider</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="SliceIdentifier"/> <xs:element ref="SliceIntentIdentifier"/> <xs:element name="VIMDescriptor"> <xs:annotation> <xs:documentation>The element that encapsulates the details of the Virtual Infrastructure Manager(s) that is assigned by a telco provider</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element name="VIM" maxOccurs="unbounded"> <xs:complexType> <xs:sequence> <xs:element ref="VIMIdentifier"/> <xs:element name="SecurityDescriptor"> <xs:annotation> <xs:documentation>The element that represents the serialized security credentials that are created by the telco provider for this VIM</xs:documentation> </xs:annotation> </xs:element> <xs:element name="Metadata" minOccurs="0"> <xs:annotation> <xs:documentation>An element that is used to represent dynamically interpretable fields in the form of key/value pairs (e.g. Location)</xs:documentation>

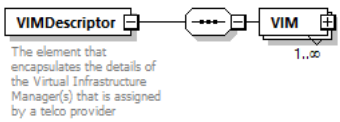
```
                    </xs:annotation>
                    <xs:complexType>
                      <xs:sequence maxOccurs="unbounded">
                        <xs:element name="Key" type="xs:string"/>
                        <xs:element name="Value" type="xs:string"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="QuotaDescriptor">
                    <xs:annotation>
                      <xs:documentation>The element that encapsulates the
Quota Availability of the VIM</xs:documentation>
                    </xs:annotation>
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="VCPUs" type="xs:int">
                          <xs:annotation>
                            <xs:documentation>The amount of
VCPUs</xs:documentation>
                          </xs:annotation>
                        </xs:element>
                        <xs:element name="Memory" type="xs:int">
                          <xs:annotation>
                            <xs:documentation>The amount of Memory in
MBytes</xs:documentation>
                          </xs:annotation>
                        </xs:element>
                        <xs:element name="Storage">
                          <xs:annotation>
                            <xs:documentation>The amount of storage in
MBytes</xs:documentation>
                          </xs:annotation>
                        </xs:element>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ComponentPlacementDescriptor">
        <xs:annotation>
          <xs:documentation>The element that encapsulates the instructions of
placement that should be performed by the Service Mesh
orchestrator</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ComponentPlacement" maxOccurs="unbounded">
              <xs:annotation>
                <xs:documentation>The element that encapsulates the details of
a component placement</xs:documentation>
              </xs:annotation>
              <xs:complexType>
```

```
                <xs:sequence>
                  <xs:element ref="VIMIdentifier"/>
                  <xs:element ref="ComponentIdentifier"/>
                  <xs:element name="InterfaceBinding" maxOccurs="unbounded">
                    <xs:annotation>
                      <xs:documentation>The element that encapsulates the
binding details of each interface</xs:documentation>
                    </xs:annotation>
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element ref="InterfaceIdentifier"/>
                        <xs:element ref="AttachmentPointIdentifier"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="QoSDescriptor">
        <xs:annotation>
          <xs:documentation>The element that encapsulates the guaranteed
QoS</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="GraphLinkQoS" minOccurs="0"
maxOccurs="unbounded">
              <xs:annotation>
                <xs:documentation>The element that encapsulates the guaranteed
QoS of Graph Links</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence>
                  <xs:element ref="GraphLinkIdentifier"/>
                  <xs:sequence>
                    <xs:element name="Delay" type="xs:int">
                      <xs:annotation>
                        <xs:documentation>The element represents the maximum
Delay that can be tolerated</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="Jitter" type="xs:int">
                      <xs:annotation>
                        <xs:documentation>The element represents the maximum
Jitter that can be tolerated</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="PacketLoss" type="xs:int">
                      <xs:annotation>
                        <xs:documentation>The element represents the maximum
Packet Loss that can be tolerated</xs:documentation>
                      </xs:annotation>
                    </xs:element>
```

```
                    <xs:element name="Throughput" type="xs:int">
                      <xs:annotation>
                        <xs:documentation>The element represents the minimum
guaranteed Throughput</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                  </xs:sequence>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="AccessQoS" maxOccurs="unbounded">
              <xs:annotation>
                <xs:documentation>The element that encapsulates the guaranteed
QoS of Access interfaces</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence>
                  <xs:element ref="InterfaceIdentifier"/>
                  <xs:element name="AccessQualityProfile">
                    <xs:annotation>
                      <xs:documentation>The element that repreents the
requested quality profile</xs:documentation>
                    </xs:annotation>
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="QCI" type="xs:string">
                          <xs:annotation>
                            <xs:documentation>The QoS class
Identifier</xs:documentation>
                          </xs:annotation>
                        </xs:element>
                        <xs:element name="ResourceType">
                          <xs:annotation>
                            <xs:documentation>The indication weather or not
Guarantted Bit Rate must be assured</xs:documentation>
                          </xs:annotation>
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:enumeration value="GBR"/>
                              <xs:enumeration value="NONGBR"/>
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:element>
                        <xs:element name="Priority" type="xs:int">
                          <xs:annotation>
                            <xs:documentation>The priority
level</xs:documentation>
                          </xs:annotation>
                        </xs:element>
                        <xs:element name="PacketDelayBudget" type="xs:int">
                          <xs:annotation>
                            <xs:documentation>The Packet Delay
Budget</xs:documentation>
                          </xs:annotation>
                        </xs:element>
                        <xs:element name="PacketErrorLossRate">
```

```
                        <xs:annotation>
                          <xs:documentation>The Packet Error Loss
Rate</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                      <xs:element name="UEType" type="xs:string">
                        <xs:annotation>
                          <xs:documentation>The Type of User
Equipment</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
```

element **Slice/VIMDescriptor**

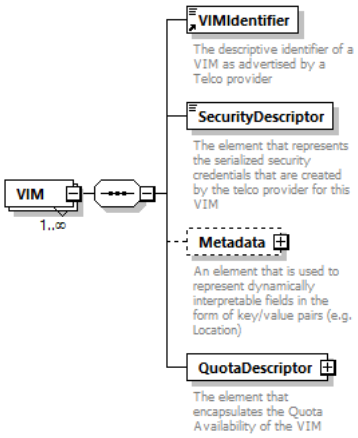
diagram	
properties	content complex
children	**VIM**
annotation	documentation The element that encapsulates the details of the Virtual Infrastructure Manager(s) that is assigned by a telco provider

source

```
<xs:element name="VIMDescriptor">
  <xs:annotation>
    <xs:documentation>The element that encapsulates the details of the Virtual
Infrastructure Manager(s) that is assigned by a telco
provider</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="VIM" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="VIMIdentifier"/>
            <xs:element name="SecurityDescriptor">
              <xs:annotation>
                <xs:documentation>The element that represents the serialized
security credentials that are created by the telco provider for this
VIM</xs:documentation>
              </xs:annotation>
            </xs:element>
```

```
            <xs:element name="Metadata" minOccurs="0">
              <xs:annotation>
                <xs:documentation>An element that is used to represent
dynamically interpretable fields in the form of key/value pairs (e.g.
Location)</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence maxOccurs="unbounded">
                  <xs:element name="Key" type="xs:string"/>
                  <xs:element name="Value" type="xs:string"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="QuotaDescriptor">
              <xs:annotation>
                <xs:documentation>The element that encapsulates the Quota
Availability of the VIM</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="VCPUs" type="xs:int">
                    <xs:annotation>
                      <xs:documentation>The amount of VCPUs</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="Memory" type="xs:int">
                    <xs:annotation>
                      <xs:documentation>The amount of Memory in
MBytes</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="Storage">
                    <xs:annotation>
                      <xs:documentation>The amount of storage in
MBytes</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

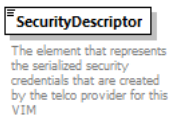
element **Slice/VIMDescriptor/VIM**

diagram	
properties	minOcc 1 maxOcc unbounded content complex
children	VIMIdentifier SecurityDescriptor Metadata QuotaDescriptor
source	<xs:element name="VIM" maxOccurs="unbounded"> <xs:complexType> <xs:sequence> <xs:element ref="VIMIdentifier"/> <xs:element name="SecurityDescriptor"> <xs:annotation> <xs:documentation>The element that represents the serialized security credentials that are created by the telco provider for this VIM</xs:documentation> </xs:annotation> </xs:element> <xs:element name="Metadata" minOccurs="0"> <xs:annotation> <xs:documentation>An element that is used to represent dynamically interpretable fields in the form of key/value pairs (e.g. Location)</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence maxOccurs="unbounded"> <xs:element name="Key" type="xs:string"/> <xs:element name="Value" type="xs:string"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="QuotaDescriptor"> <xs:annotation> <xs:documentation>The element that encapsulates the Quota Availability of the VIM</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element name="VCPUs" type="xs:int"> <xs:annotation> <xs:documentation>The amount of VCPUs</xs:documentation> </xs:annotation> </xs:element>

```
            <xs:element name="Memory" type="xs:int">
              <xs:annotation>
                <xs:documentation>The amount of Memory in
MBytes</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="Storage">
              <xs:annotation>
                <xs:documentation>The amount of storage in
MBytes</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **Slice/VIMDescriptor/VIM/SecurityDescriptor**

diagram



annotation

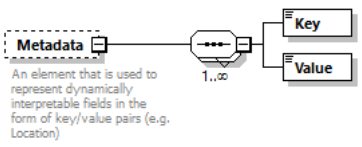
documentation
The element that represents the serialized security credentials that are created by the telco provider for this VIM

source

```
<xs:element name="SecurityDescriptor">
  <xs:annotation>
    <xs:documentation>The element that represents the serialized security
credentials that are created by the telco provider for this
VIM</xs:documentation>
  </xs:annotation>
</xs:element>
```

element **Slice/VIMDescriptor/VIM/Metadata**

diagram



properties

minOcc	0
maxOcc	1
content	complex

children **Key Value**

annotation

documentation
An element that is used to represent dynamically interpretable fields in the form of key/value pairs (e.g. Location)

source

```
<xs:element name="Metadata" minOccurs="0">
  <xs:annotation>
    <xs:documentation>An element that is used to represent dynamically
interpretable fields in the form of key/value pairs (e.g.
Location)</xs:documentation>
```

```
  </xs:annotation>
  <xs:complexType>
    <xs:sequence maxOccurs="unbounded">
      <xs:element name="Key" type="xs:string"/>
      <xs:element name="Value" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

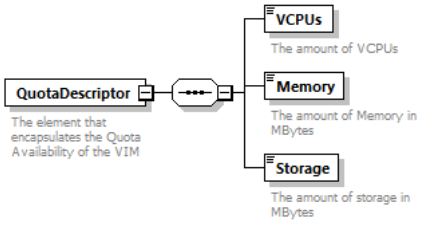
element **Slice/VIMDescriptor/VIM/Metadata/Key**

diagram	
type	**xs:string**
properties	content simple
source	`<xs:element name="Key" type="xs:string"/>`

element **Slice/VIMDescriptor/VIM/Metadata/Value**

diagram	
type	**xs:string**
properties	content simple
source	`<xs:element name="Value" type="xs:string"/>`

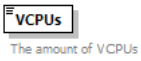
element **Slice/VIMDescriptor/VIM/QuotaDescriptor**

diagram	
properties	content complex
children	**VCPUs Memory Storage**
annotation	documentation The element that encapsulates the Quota Availability of the VIM

source	`<xs:element name="QuotaDescriptor">`
	`  <xs:annotation>`
	`  <xs:documentation>The element that encapsulates the Quota Availability of the VIM</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element name="VCPUs" type="xs:int">`
	`  <xs:annotation>`
	`  <xs:documentation>The amount of VCPUs</xs:documentation>`

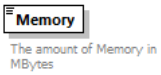
```
        </xs:annotation>
      </xs:element>
      <xs:element name="Memory" type="xs:int">
        <xs:annotation>
          <xs:documentation>The amount of Memory in MBytes</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Storage">
        <xs:annotation>
          <xs:documentation>The amount of storage in MBytes</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **Slice/VIMDescriptor/VIM/QuotaDescriptor/VCPUs**

diagram	
type	**xs:int**
properties	content simple
annotation	documentation The amount of VCPUs

source	`<xs:element name="VCPUs" type="xs:int">`
	`  <xs:annotation>`
	`  <xs:documentation>The amount of VCPUs</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **Slice/VIMDescriptor/VIM/QuotaDescriptor/Memory**

diagram	
type	**xs:int**
properties	content simple
annotation	documentation The amount of Memory in MBytes

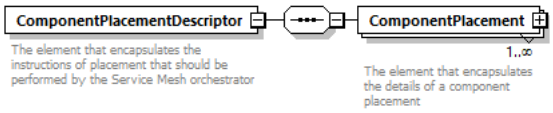
source	`<xs:element name="Memory" type="xs:int">`
	`  <xs:annotation>`
	`  <xs:documentation>The amount of Memory in MBytes</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **Slice/VIMDescriptor/VIM/QuotaDescriptor/Storage**

diagram	

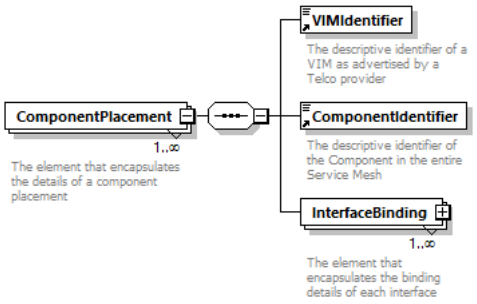
annotation	documentation The amount of storage in MBytes
source	<xs:element name="Storage"> <xs:annotation> <xs:documentation>The amount of storage in MBytes</xs:documentation> </xs:annotation> </xs:element>

element **Slice/ComponentPlacementDescriptor**

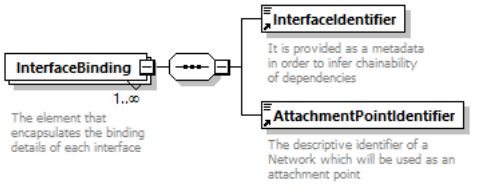
diagram	
properties	content complex
children	**ComponentPlacement**
annotation	documentation The element that encapsulates the instructions of placement that should be performed by the Service Mesh orchestrator
source	<xs:element name="ComponentPlacementDescriptor"> <xs:annotation> <xs:documentation>The element that encapsulates the instructions of placement that should be performed by the Service Mesh orchestrator</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element name="ComponentPlacement" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>The element that encapsulates the details of a component placement</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="VIMIdentifier"/> <xs:element ref="ComponentIdentifier"/> <xs:element name="InterfaceBinding" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>The element that encapsulates the binding details of each interface</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="InterfaceIdentifier"/> <xs:element ref="AttachmentPointIdentifier"/> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType>

	</xs:element>

element **Slice/ComponentPlacementDescriptor/ComponentPlacement**

diagram	
properties	minOcc 1 maxOcc unbounded content complex
children	VIMIdentifier ComponentIdentifier InterfaceBinding
annotation	documentation The element that encapsulates the details of a component placement
source	<xs:element name="ComponentPlacement" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>The element that encapsulates the details of a component placement</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="VIMIdentifier"/> <xs:element ref="ComponentIdentifier"/> <xs:element name="InterfaceBinding" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>The element that encapsulates the binding details of each interface</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="InterfaceIdentifier"/> <xs:element ref="AttachmentPointIdentifier"/> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </xs:element>

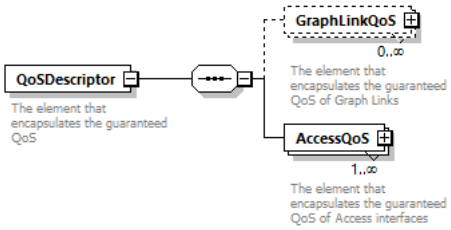
element **Slice/ComponentPlacementDescriptor/ComponentPlacement/InterfaceBinding**

diagram	

properties	minOcc 1 maxOcc unbounded content complex
children	**InterfaceIdentifier AttachmentPointIdentifier**
annotation	documentation The element that encapsulates the binding details of each interface

source	`<xs:element name="InterfaceBinding" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation>The element that encapsulates the binding details of each interface</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element ref="InterfaceIdentifier"/>`
	`  <xs:element ref="AttachmentPointIdentifier"/>`
	`  </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

element **Slice/QoSDescriptor**

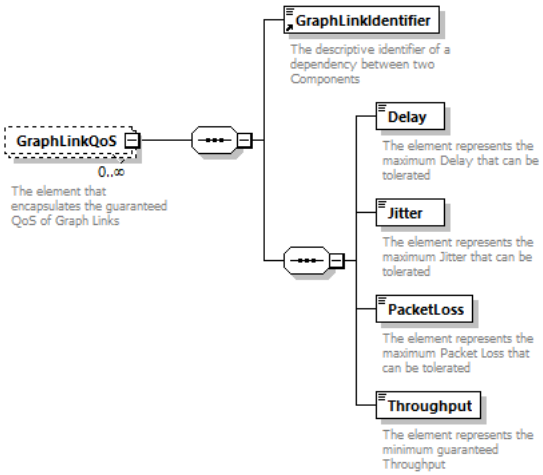
diagram	
properties	content complex
children	**GraphLinkQoS AccessQoS**
annotation	documentation The element that encapsulates the guaranteed QoS

source	`<xs:element name="QoSDescriptor">`
	`  <xs:annotation>`
	`  <xs:documentation>The element that encapsulates the guaranteed QoS</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element name="GraphLinkQoS" minOccurs="0" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation>The element that encapsulates the guaranteed QoS of Graph Links</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element ref="GraphLinkIdentifier"/>`
	`  <xs:sequence>`
	`  <xs:element name="Delay" type="xs:int">`
	`  <xs:annotation>`
	`  <xs:documentation>The element represents the maximum Delay that can be tolerated</xs:documentation>`

```
                </xs:annotation>
              </xs:element>
              <xs:element name="Jitter" type="xs:int">
                <xs:annotation>
                  <xs:documentation>The element represents the maximum Jitter
that can be tolerated</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="PacketLoss" type="xs:int">
                <xs:annotation>
                  <xs:documentation>The element represents the maximum Packet
Loss that can be tolerated</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="Throughput" type="xs:int">
                <xs:annotation>
                  <xs:documentation>The element represents the minimum
guaranteed Throughput</xs:documentation>
                </xs:annotation>
              </xs:element>
            </xs:sequence>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="AccessQoS" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>The element that encapsulates the guaranteed QoS
of Access interfaces</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="InterfaceIdentifier"/>
            <xs:element name="AccessQualityProfile">
              <xs:annotation>
                <xs:documentation>The element that repreents the requested
quality profile</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="QCI" type="xs:string">
                    <xs:annotation>
                      <xs:documentation>The QoS class
Identifier</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="ResourceType">
                    <xs:annotation>
                      <xs:documentation>The indication weather or not
Guarantted Bit Rate must be assured</xs:documentation>
                    </xs:annotation>
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:enumeration value="GBR"/>
                        <xs:enumeration value="NONGBR"/>
                      </xs:restriction>
                    </xs:simpleType>
```

```
                    </xs:element>
                    <xs:element name="Priority" type="xs:int">
                      <xs:annotation>
                        <xs:documentation>The priority level</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="PacketDelayBudget" type="xs:int">
                      <xs:annotation>
                        <xs:documentation>The Packet Delay
Budget</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="PacketErrorLossRate">
                      <xs:annotation>
                        <xs:documentation>The Packet Error Loss
Rate</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="UEType" type="xs:string">
                      <xs:annotation>
                        <xs:documentation>The Type of User
Equipment</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

element **Slice/QoSDescriptor/GraphLinkQoS**

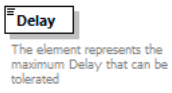
diagram	
properties	minOcc 0 maxOcc unbounded content complex

children	GraphLinkIdentifier Delay Jitter PacketLoss Throughput
annotation	documentation The element that encapsulates the guaranteed QoS of Graph Links

source

```
<xs:element name="GraphLinkQoS" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>The element that encapsulates the guaranteed QoS of
Graph Links</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="GraphLinkIdentifier"/>
      <xs:sequence>
        <xs:element name="Delay" type="xs:int">
          <xs:annotation>
            <xs:documentation>The element represents the maximum Delay that
can be tolerated</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Jitter" type="xs:int">
          <xs:annotation>
            <xs:documentation>The element represents the maximum Jitter that
can be tolerated</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="PacketLoss" type="xs:int">
          <xs:annotation>
            <xs:documentation>The element represents the maximum Packet Loss
that can be tolerated</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Throughput" type="xs:int">
          <xs:annotation>
            <xs:documentation>The element represents the minimum guaranteed
Throughput</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **Slice/QoSDescriptor/GraphLinkQoS/Delay**

diagram	
type	**xs:int**
properties	content simple
annotation	documentation The element represents the maximum Delay that can be tolerated

source

```
<xs:element name="Delay" type="xs:int">
  <xs:annotation>
    <xs:documentation>The element represents the maximum Delay that can be
```

	tolerated</xs:documentation> </xs:annotation> </xs:element>

element **Slice/QoSDescriptor/GraphLinkQoS/Jitter**

diagram	
type	**xs:int**
properties	content simple
annotation	documentation The element represents the maximum Jitter that can be tolerated
source	<xs:element name="Jitter" type="xs:int"> <xs:annotation> <xs:documentation>The element represents the maximum Jitter that can be tolerated</xs:documentation> </xs:annotation> </xs:element>

element **Slice/QoSDescriptor/GraphLinkQoS/PacketLoss**

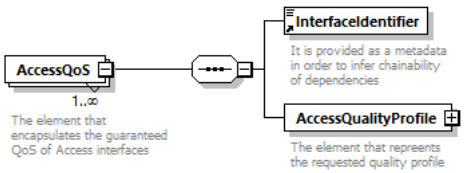
diagram	
type	**xs:int**
properties	content simple
annotation	documentation The element represents the maximum Packet Loss that can be tolerated
source	<xs:element name="PacketLoss" type="xs:int"> <xs:annotation> <xs:documentation>The element represents the maximum Packet Loss that can be tolerated</xs:documentation> </xs:annotation> </xs:element>

element **Slice/QoSDescriptor/GraphLinkQoS/Throughput**

diagram	
type	**xs:int**
properties	content simple
annotation	documentation The element represents the minimum guaranteed Throughput
source	<xs:element name="Throughput" type="xs:int"> <xs:annotation>

```
    <xs:documentation>The element represents the minimum guaranteed
Throughput</xs:documentation>
  </xs:annotation>
</xs:element>
```

element **Slice/QoSDescriptor/AccessQoS**

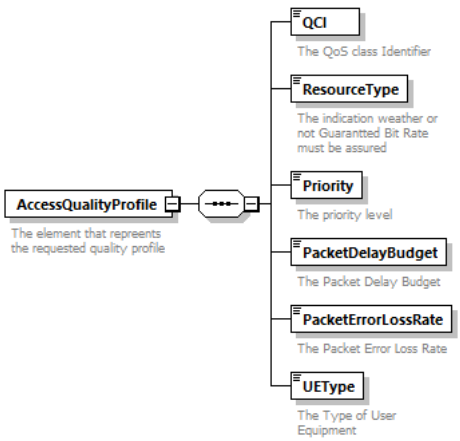
diagram	
properties	minOcc 1 maxOcc unbounded content complex
children	**InterfaceIdentifier** **AccessQualityProfile**
annotation	documentation The element that encapsulates the guaranteed QoS of Access interfaces

source

```
<xs:element name="AccessQoS" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>The element that encapsulates the guaranteed QoS of
Access interfaces</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="InterfaceIdentifier"/>
      <xs:element name="AccessQualityProfile">
        <xs:annotation>
          <xs:documentation>The element that repreents the requested quality
profile</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="QCI" type="xs:string">
              <xs:annotation>
                <xs:documentation>The QoS class Identifier</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="ResourceType">
              <xs:annotation>
                <xs:documentation>The indication weather or not Guarantted Bit
Rate must be assured</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="GBR"/>
                  <xs:enumeration value="NONGBR"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="Priority" type="xs:int">
              <xs:annotation>
                <xs:documentation>The priority level</xs:documentation>
              </xs:annotation>
```

```
            </xs:element>
            <xs:element name="PacketDelayBudget" type="xs:int">
              <xs:annotation>
                <xs:documentation>The Packet Delay Budget</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="PacketErrorLossRate">
              <xs:annotation>
                <xs:documentation>The Packet Error Loss
Rate</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="UEType" type="xs:string">
              <xs:annotation>
                <xs:documentation>The Type of User
Equipment</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

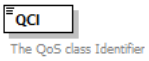
element **Slice/QoSDescriptor/AccessQoS/AccessQualityProfile**

diagram	
properties	content complex
children	QCI ResourceType Priority PacketDelayBudget PacketErrorLossRate UEType
annotation	documentation The element that repreents the requested quality profile

source	`<xs:element name="AccessQualityProfile">`
	`  <xs:annotation>`
	`  <xs:documentation>The element that repreents the requested quality profile</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element name="QCI" type="xs:string">`

```
        <xs:annotation>
          <xs:documentation>The QoS class Identifier</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="ResourceType">
        <xs:annotation>
          <xs:documentation>The indication weather or not Guarantted Bit Rate
must be assured</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="GBR"/>
            <xs:enumeration value="NONGBR"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Priority" type="xs:int">
        <xs:annotation>
          <xs:documentation>The priority level</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="PacketDelayBudget" type="xs:int">
        <xs:annotation>
          <xs:documentation>The Packet Delay Budget</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="PacketErrorLossRate">
        <xs:annotation>
          <xs:documentation>The Packet Error Loss Rate</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="UEType" type="xs:string">
        <xs:annotation>
          <xs:documentation>The Type of User Equipment</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **Slice/QoSDescriptor/AccessQoS/AccessQualityProfile/QCI**

diagram	
type	**xs:string**
properties	content simple
annotation	documentation The QoS class Identifier

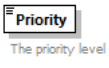
source	`<xs:element name="QCI" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation>The QoS class Identifier</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **Slice/QoSDescriptor/AccessQoS/AccessQualityProfile/ResourceType**

diagram	
type	restriction of **xs:string**
properties	content simple
facets	Kind Value Annotation enumeration GBR enumeration NONGBR
annotation	documentation The indication weather or not Guarantted Bit Rate must be assured

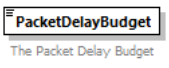
source	`<xs:element name="ResourceType">`
	`<xs:annotation>`
	`<xs:documentation>`The indication weather or not Guarantted Bit Rate must be assured`</xs:documentation>`
	`</xs:annotation>`
	`<xs:simpleType>`
	`<xs:restriction base="xs:string">`
	`<xs:enumeration value="GBR"/>`
	`<xs:enumeration value="NONGBR"/>`
	`</xs:restriction>`
	`</xs:simpleType>`
	`</xs:element>`

element **Slice/QoSDescriptor/AccessQoS/AccessQualityProfile/Priority**

diagram	
type	**xs:int**
properties	content simple
annotation	documentation The priority level

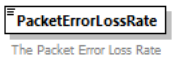
source	`<xs:element name="Priority" type="xs:int">`
	`<xs:annotation>`
	`<xs:documentation>`The priority level`</xs:documentation>`
	`</xs:annotation>`
	`</xs:element>`

element **Slice/QoSDescriptor/AccessQoS/AccessQualityProfile/PacketDelayBudget**

diagram	
type	**xs:int**
properties	content simple
annotation	documentation The Packet Delay Budget

source	<xs:element name="PacketDelayBudget" type="xs:int"> <xs:annotation> <xs:documentation>The Packet Delay Budget</xs:documentation> </xs:annotation> </xs:element>

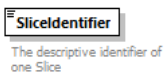
element **Slice/QoSDescriptor/AccessQoS/AccessQualityProfile/PacketErrorLossRate**

diagram	
annotation	documentation The Packet Error Loss Rate
source	<xs:element name="PacketErrorLossRate"> <xs:annotation> <xs:documentation>The Packet Error Loss Rate</xs:documentation> </xs:annotation> </xs:element>

element **Slice/QoSDescriptor/AccessQoS/AccessQualityProfile/UEType**

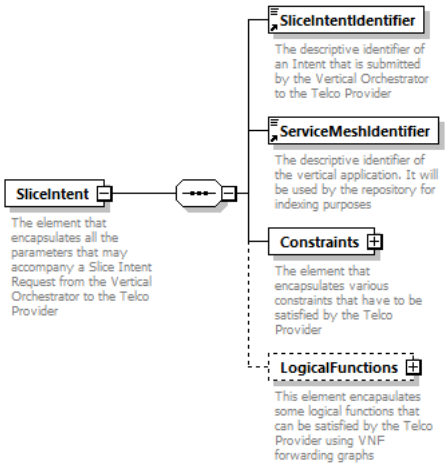
diagram	
type	**xs:string**
properties	content simple
annotation	documentation The Type of User Equipment
source	<xs:element name="UEType" type="xs:string"> <xs:annotation> <xs:documentation>The Type of User Equipment</xs:documentation> </xs:annotation> </xs:element>

element **SliceIdentifier**

diagram	
type	**xs:string**
properties	content simple
used by	element **Slice**
annotation	documentation The descriptive identifier of one Slice
source	<xs:element name="SliceIdentifier" type="xs:string"> <xs:annotation> <xs:documentation>The descriptive identifier of one Slice</xs:documentation> </xs:annotation>

	</xs:element>

element **SliceIntent**

diagram	
properties	content complex
children	SliceIntentIdentifier ServiceMeshIdentifier Constraints LogicalFunctions
annotation	documentation The element that encapsulates all the parameters that may accompany a Slice Intent Request from the Vertical Orchestrator to the Telco Provider
source	<xs:element name="SliceIntent"> <xs:annotation> <xs:documentation>The element that encapsulates all the parameters that may accompany a Slice Intent Request from the Vertical Orchestrator to the Telco Provider</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="SliceIntentIdentifier"> <xs:annotation> <xs:documentation>The descriptive identifier of an Intent that is submitted by the Vertical Orchestrator to the Telco Provider</xs:documentation> </xs:annotation> </xs:element> <xs:element ref="ServiceMeshIdentifier"/> <xs:element name="Constraints"> <xs:annotation> <xs:documentation>The element that encapsulates various constraints that have to be satisfied by the Telco Provider</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element name="ComponentHostingConstraints" minOccurs="0"> <xs:annotation> <xs:documentation>It represents the constraints that relate to the Component hosting requirements</xs:documentation> </xs:annotation> <xs:complexType>

```
                <xs:sequence>
                  <xs:element name="ResourceConstraint" minOccurs="0"
maxOccurs="unbounded">
                    <xs:annotation>
                      <xs:documentation>The element that encapsulates the
Resource constraints. Through this element the Vertical orhestrator may
require specific quota of VCPU, Memory and Storage</xs:documentation>
                    </xs:annotation>
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element ref="RequirementIdentifier"/>
                        <xs:element ref="ComponentIdentifier"/>
                        <xs:element name="ConstraintType">
                          <xs:annotation>
                            <xs:documentation>The Type of the
Constraint</xs:documentation>
                          </xs:annotation>
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:enumeration value="HARD"/>
                              <xs:enumeration value="SOFT"/>
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:element>
                        <xs:choice>
                          <xs:element name="VCPU" type="xs:int">
                            <xs:annotation>
                              <xs:documentation>The amount of
VCPU</xs:documentation>
                            </xs:annotation>
                          </xs:element>
                          <xs:element name="RAM" type="xs:int">
                            <xs:annotation>
                              <xs:documentation>The amount of RAM in Mega
Bytes</xs:documentation>
                            </xs:annotation>
                          </xs:element>
                          <xs:element name="Storage" type="xs:int">
                            <xs:annotation>
                              <xs:documentation>The amount of storage in Mega
Bytes</xs:documentation>
                            </xs:annotation>
                          </xs:element>
                        </xs:choice>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="LocationConstraint" minOccurs="0"
maxOccurs="unbounded">
                    <xs:annotation>
                      <xs:documentation>The element that encapsulates the
location-related constraints</xs:documentation>
                    </xs:annotation>
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element ref="RequirementIdentifier"/>
```

```
                        <xs:element ref="ComponentIdentifier"/>
                        <xs:element name="ConstraintType">
                          <xs:annotation>
                            <xs:documentation>The Type of the
Constraint</xs:documentation>
                          </xs:annotation>
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:enumeration value="HARD"/>
                              <xs:enumeration value="SOFT"/>
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:element>
                        <xs:element name="Location" type="xs:string">
                          <xs:annotation>
                            <xs:documentation>The location
descriptor</xs:documentation>
                          </xs:annotation>
                        </xs:element>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="GraphLinkConstraints" minOccurs="0">
              <xs:annotation>
                <xs:documentation>It represents the constraints that relate to
the link between two Components of the Service Mesh</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="GraphLinkQoSConstraint" minOccurs="0"
maxOccurs="unbounded">
                    <xs:annotation>
                      <xs:documentation>The element that encapsulates the
graph-link constraints</xs:documentation>
                    </xs:annotation>
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element ref="RequirementIdentifier"/>
                        <xs:element ref="GraphLinkIdentifier"/>
                        <xs:element name="ConstraintType">
                          <xs:annotation>
                            <xs:documentation>The Type of the
Constraint</xs:documentation>
                          </xs:annotation>
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:enumeration value="HARD"/>
                              <xs:enumeration value="SOFT"/>
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:element>
                        <xs:choice>
                          <xs:element name="Delay" type="xs:int">
```

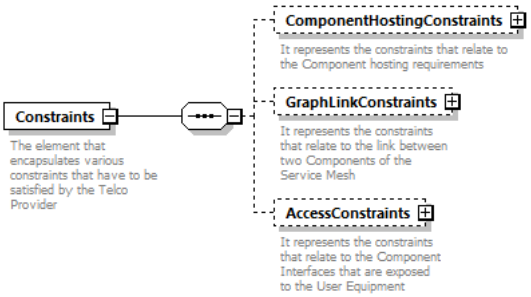
```
                              <xs:annotation>
                                <xs:documentation>The element represents the
maximum Delay that can be tolerated</xs:documentation>
                              </xs:annotation>
                            </xs:element>
                            <xs:element name="Jitter" type="xs:int">
                              <xs:annotation>
                                <xs:documentation>The element represents the
maximum Jitter that can be tolerated</xs:documentation>
                              </xs:annotation>
                            </xs:element>
                            <xs:element name="PacketLoss" type="xs:int">
                              <xs:annotation>
                                <xs:documentation>The element represents the
maximum Packet Loss that can be tolerated</xs:documentation>
                              </xs:annotation>
                            </xs:element>
                            <xs:element name="Throughput" type="xs:int">
                              <xs:annotation>
                                <xs:documentation>The element represents the
minimum guaranteed Throughput</xs:documentation>
                              </xs:annotation>
                            </xs:element>
                          </xs:choice>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="AccessConstraints" minOccurs="0">
                <xs:annotation>
                  <xs:documentation>It represents the constraints that relate to
the Component Interfaces that are exposed to the User
Equipment</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="AccessConstraint" minOccurs="0"
maxOccurs="unbounded">
                      <xs:annotation>
                        <xs:documentation>The element that encapsulates the
constraints that relate with the communication of UEs with the Component
Interfaces</xs:documentation>
                      </xs:annotation>
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element ref="RequirementIdentifier"/>
                          <xs:element ref="InterfaceIdentifier"/>
                          <xs:element name="ConstraintType">
                            <xs:annotation>
                              <xs:documentation>The Type of the
Constraint</xs:documentation>
                            </xs:annotation>
                            <xs:simpleType>
                              <xs:restriction base="xs:string">
```

```
                                <xs:enumeration value="HARD"/>
                                <xs:enumeration value="SOFT"/>
                              </xs:restriction>
                            </xs:simpleType>
                          </xs:element>
                          <xs:element name="AccessQualityProfile">
                            <xs:annotation>
                              <xs:documentation>The element that repreents the
requested quality profile</xs:documentation>
                            </xs:annotation>
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="QCI" type="xs:string">
                                  <xs:annotation>
                                    <xs:documentation>The QoS class
Identifier</xs:documentation>
                                  </xs:annotation>
                                </xs:element>
                                <xs:element name="ResourceType">
                                  <xs:annotation>
                                    <xs:documentation>The indication weather or
not Guarantted Bit Rate must be assured</xs:documentation>
                                  </xs:annotation>
                                  <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                      <xs:enumeration value="GBR"/>
                                      <xs:enumeration value="NONGBR"/>
                                    </xs:restriction>
                                  </xs:simpleType>
                                </xs:element>
                                <xs:element name="Priority" type="xs:int">
                                  <xs:annotation>
                                    <xs:documentation>The priority
level</xs:documentation>
                                  </xs:annotation>
                                </xs:element>
                                <xs:element name="PacketDelayBudget"
type="xs:int">
                                  <xs:annotation>
                                    <xs:documentation>The Packet Delay
Budget</xs:documentation>
                                  </xs:annotation>
                                </xs:element>
                                <xs:element name="PacketErrorLossRate">
                                  <xs:annotation>
                                    <xs:documentation>The Packet Error Loss
Rate</xs:documentation>
                                  </xs:annotation>
                                </xs:element>
                                <xs:element name="UEType" type="xs:string">
                                  <xs:annotation>
                                    <xs:documentation>The Type of User
Equipment</xs:documentation>
                                  </xs:annotation>
                                </xs:element>
                              </xs:sequence>
```

```
                        </xs:complexType>
                      </xs:element>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="LogicalFunctions" minOccurs="0">
      <xs:annotation>
        <xs:documentation>This element encapaulates some logical functions
that can be satisfied by the Telco Provider using VNF forwarding
graphs</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Firewalling" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="FirewallConfiguration" minOccurs="0"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="VPN" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="VPNConfiguration" minOccurs="0"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="LawfulInspection" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="LawfulInspectionConfiguration"
minOccurs="0" maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="IDS_IPS" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="IDSIPSConfiguration" minOccurs="0"
maxOccurs="unbounded"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

	</xs:element>

element **SliceIntent/Constraints**

diagram	
properties	content complex
children	ComponentHostingConstraints GraphLinkConstraints AccessConstraints
annotation	documentation The element that encapsulates various constraints that have to be satisfied by the Telco Provider
source	<xs:element name="Constraints"> <xs:annotation> <xs:documentation>The element that encapsulates various constraints that have to be satisfied by the Telco Provider</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element name="ComponentHostingConstraints" minOccurs="0"> <xs:annotation> <xs:documentation>It represents the constraints that relate to the Component hosting requirements</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element name="ResourceConstraint" minOccurs="0" maxOccurs="unbounded"> <xs:annotation> <xs:documentation>The element that encapsulates the Resource constraints. Through this element the Vertical orhestrator may require specific quota of VCPU, Memory and Storage</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element ref="RequirementIdentifier"/> <xs:element ref="ComponentIdentifier"/> <xs:element name="ConstraintType"> <xs:annotation> <xs:documentation>The Type of the Constraint</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="HARD"/> <xs:enumeration value="SOFT"/> </xs:restriction> </xs:simpleType>

```
                    </xs:element>
                    <xs:choice>
                      <xs:element name="VCPU" type="xs:int">
                        <xs:annotation>
                          <xs:documentation>The amount of
VCPU</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                      <xs:element name="RAM" type="xs:int">
                        <xs:annotation>
                          <xs:documentation>The amount of RAM in Mega
Bytes</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                      <xs:element name="Storage" type="xs:int">
                        <xs:annotation>
                          <xs:documentation>The amount of storage in Mega
Bytes</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                    </xs:choice>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="LocationConstraint" minOccurs="0"
maxOccurs="unbounded">
                <xs:annotation>
                  <xs:documentation>The element that encapsulates the location-
related constraints</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                  <xs:sequence>
                    <xs:element ref="RequirementIdentifier"/>
                    <xs:element ref="ComponentIdentifier"/>
                    <xs:element name="ConstraintType">
                      <xs:annotation>
                        <xs:documentation>The Type of the
Constraint</xs:documentation>
                      </xs:annotation>
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:enumeration value="HARD"/>
                          <xs:enumeration value="SOFT"/>
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                    <xs:element name="Location" type="xs:string">
                      <xs:annotation>
                        <xs:documentation>The location
descriptor</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
```

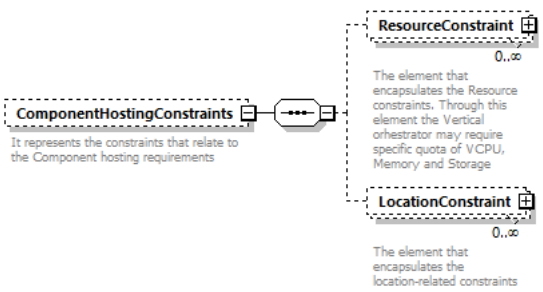
```
        </xs:complexType>
      </xs:element>
      <xs:element name="GraphLinkConstraints" minOccurs="0">
        <xs:annotation>
          <xs:documentation>It represents the constraints that relate to the
link between two Components of the Service Mesh</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="GraphLinkQoSConstraint" minOccurs="0"
maxOccurs="unbounded">
              <xs:annotation>
                <xs:documentation>The element that encapsulates the graph-link
constraints</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence>
                  <xs:element ref="RequirementIdentifier"/>
                  <xs:element ref="GraphLinkIdentifier"/>
                  <xs:element name="ConstraintType">
                    <xs:annotation>
                      <xs:documentation>The Type of the
Constraint</xs:documentation>
                    </xs:annotation>
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:enumeration value="HARD"/>
                        <xs:enumeration value="SOFT"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:choice>
                    <xs:element name="Delay" type="xs:int">
                      <xs:annotation>
                        <xs:documentation>The element represents the maximum
Delay that can be tolerated</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="Jitter" type="xs:int">
                      <xs:annotation>
                        <xs:documentation>The element represents the maximum
Jitter that can be tolerated</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="PacketLoss" type="xs:int">
                      <xs:annotation>
                        <xs:documentation>The element represents the maximum
Packet Loss that can be tolerated</xs:documentation>
                      </xs:annotation>
                    </xs:element>
                    <xs:element name="Throughput" type="xs:int">
                      <xs:annotation>
                        <xs:documentation>The element represents the minimum
guaranteed Throughput</xs:documentation>
                      </xs:annotation>
                    </xs:element>
```

```
                </xs:choice>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="AccessConstraints" minOccurs="0">
      <xs:annotation>
        <xs:documentation>It represents the constraints that relate to the
Component Interfaces that are exposed to the User Equipment</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="AccessConstraint" minOccurs="0"
maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>The element that encapsulates the
constraints that relate with the communication of UEs with the Component
Interfaces</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="RequirementIdentifier"/>
                <xs:element ref="InterfaceIdentifier"/>
                <xs:element name="ConstraintType">
                  <xs:annotation>
                    <xs:documentation>The Type of the
Constraint</xs:documentation>
                  </xs:annotation>
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:enumeration value="HARD"/>
                      <xs:enumeration value="SOFT"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
                <xs:element name="AccessQualityProfile">
                  <xs:annotation>
                    <xs:documentation>The element that repreents the
requested quality profile</xs:documentation>
                  </xs:annotation>
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="QCI" type="xs:string">
                        <xs:annotation>
                          <xs:documentation>The QoS class
Identifier</xs:documentation>
                        </xs:annotation>
                      </xs:element>
                      <xs:element name="ResourceType">
                        <xs:annotation>
                          <xs:documentation>The indication weather or not
Guarantted Bit Rate must be assured</xs:documentation>
                        </xs:annotation>
                        <xs:simpleType>
```

```
                              <xs:restriction base="xs:string">
                                <xs:enumeration value="GBR"/>
                                <xs:enumeration value="NONGBR"/>
                              </xs:restriction>
                            </xs:simpleType>
                          </xs:element>
                          <xs:element name="Priority" type="xs:int">
                            <xs:annotation>
                              <xs:documentation>The priority
level</xs:documentation>
                            </xs:annotation>
                          </xs:element>
                          <xs:element name="PacketDelayBudget" type="xs:int">
                            <xs:annotation>
                              <xs:documentation>The Packet Delay
Budget</xs:documentation>
                            </xs:annotation>
                          </xs:element>
                          <xs:element name="PacketErrorLossRate">
                            <xs:annotation>
                              <xs:documentation>The Packet Error Loss
Rate</xs:documentation>
                            </xs:annotation>
                          </xs:element>
                          <xs:element name="UEType" type="xs:string">
                            <xs:annotation>
                              <xs:documentation>The Type of User
Equipment</xs:documentation>
                            </xs:annotation>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

element **SliceIntent/Constraints/ComponentHostingConstraints**

diagram



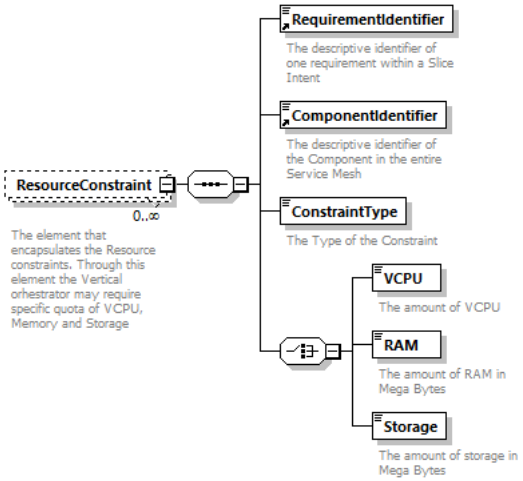
properties	minOcc 0 maxOcc 1 content complex
children	**ResourceConstraint** **LocationConstraint**
annotation	documentation It represents the constraints that relate to the Component hosting requirements
source	

```
<xs:element name="ComponentHostingConstraints" minOccurs="0">
  <xs:annotation>
    <xs:documentation>It represents the constraints that relate to the
Component hosting requirements</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ResourceConstraint" minOccurs="0"
maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>The element that encapsulates the Resource
constraints. Through this element the Vertical orhestrator may require
specific quota of VCPU, Memory and Storage</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="RequirementIdentifier"/>
            <xs:element ref="ComponentIdentifier"/>
            <xs:element name="ConstraintType">
              <xs:annotation>
                <xs:documentation>The Type of the
Constraint</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="HARD"/>
                  <xs:enumeration value="SOFT"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:choice>
              <xs:element name="VCPU" type="xs:int">
                <xs:annotation>
                  <xs:documentation>The amount of VCPU</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="RAM" type="xs:int">
                <xs:annotation>
                  <xs:documentation>The amount of RAM in Mega
Bytes</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="Storage" type="xs:int">
                <xs:annotation>
                  <xs:documentation>The amount of storage in Mega
Bytes</xs:documentation>
                </xs:annotation>
              </xs:element>
            </xs:choice>
          </xs:sequence>
```

```
        </xs:complexType>
      </xs:element>
      <xs:element name="LocationConstraint" minOccurs="0"
maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>The element that encapsulates the location-related
constraints</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="RequirementIdentifier"/>
            <xs:element ref="ComponentIdentifier"/>
            <xs:element name="ConstraintType">
              <xs:annotation>
                <xs:documentation>The Type of the
Constraint</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="HARD"/>
                  <xs:enumeration value="SOFT"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="Location" type="xs:string">
              <xs:annotation>
                <xs:documentation>The location descriptor</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **SliceIntent/Constraints/ComponentHostingConstraints/ResourceConstraint**

diagram



properties

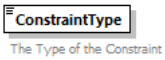
minOcc 0

	maxOcc unbounded content complex
children	**RequirementIdentifier ComponentIdentifier ConstraintType VCPU RAM Storage**
annotation	documentation The element that encapsulates the Resource constraints. Through this element the Vertical orhestrator may require specific quota of VCPU, Memory and Storage

source

```
<xs:element name="ResourceConstraint" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>The element that encapsulates the Resource constraints.
Through this element the Vertical orhestrator may require specific quota of
VCPU, Memory and Storage</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="RequirementIdentifier"/>
      <xs:element ref="ComponentIdentifier"/>
      <xs:element name="ConstraintType">
        <xs:annotation>
          <xs:documentation>The Type of the Constraint</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="HARD"/>
            <xs:enumeration value="SOFT"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:choice>
        <xs:element name="VCPU" type="xs:int">
          <xs:annotation>
            <xs:documentation>The amount of VCPU</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="RAM" type="xs:int">
          <xs:annotation>
            <xs:documentation>The amount of RAM in Mega
Bytes</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Storage" type="xs:int">
          <xs:annotation>
            <xs:documentation>The amount of storage in Mega
Bytes</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **SliceIntent/Constraints/ComponentHostingConstraints/ResourceConstraint/ConstraintType**

diagram	

type	restriction of **xs:string**
properties	content simple
facets	Kind Value Annotation enumeration HARD enumeration SOFT
annotation	documentation The Type of the Constraint

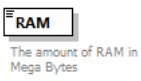
source	`<xs:element name="ConstraintType">`
	`  <xs:annotation>`
	`  <xs:documentation>The Type of the Constraint</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:simpleType>`
	`  <xs:restriction base="xs:string">`
	`  <xs:enumeration value="HARD"/>`
	`  <xs:enumeration value="SOFT"/>`
	`  </xs:restriction>`
	`  </xs:simpleType>`
	`</xs:element>`

element **SliceIntent/Constraints/ComponentHostingConstraints/ResourceConstraint/VCPU**

diagram	
type	**xs:int**
properties	content simple
annotation	documentation The amount of VCPU

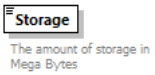
source	`<xs:element name="VCPU" type="xs:int">`
	`  <xs:annotation>`
	`  <xs:documentation>The amount of VCPU</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **SliceIntent/Constraints/ComponentHostingConstraints/ResourceConstraint/RAM**

diagram	
type	**xs:int**
properties	content simple
annotation	documentation The amount of RAM in Mega Bytes

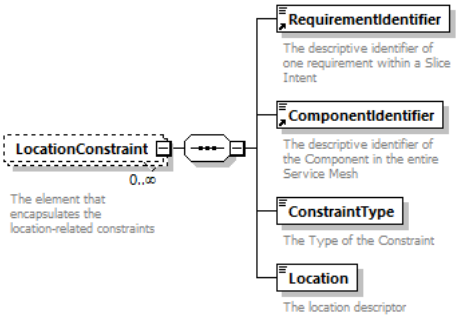
source	`<xs:element name="RAM" type="xs:int">`
	`  <xs:annotation>`
	`  <xs:documentation>The amount of RAM in Mega Bytes</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **SliceIntent/Constraints/ComponentHostingConstraints/ResourceConstraint/Storage**

diagram	
type	**xs:int**
properties	content simple
annotation	documentation The amount of storage in Mega Bytes

source	`<xs:element name="Storage" type="xs:int">`
	`  <xs:annotation>`
	`  <xs:documentation>The amount of storage in Mega Bytes</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

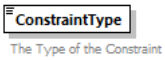
element **SliceIntent/Constraints/ComponentHostingConstraints/LocationConstraint**

diagram	
properties	minOcc 0 maxOcc unbounded content complex
children	**RequirementIdentifier ComponentIdentifier ConstraintType Location**
annotation	documentation The element that encapsulates the location-related constraints

source	`<xs:element name="LocationConstraint" minOccurs="0" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation>The element that encapsulates the location-related constraints</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element ref="RequirementIdentifier"/>`
	`  <xs:element ref="ComponentIdentifier"/>`
	`  <xs:element name="ConstraintType">`
	`  <xs:annotation>`
	`  <xs:documentation>The Type of the Constraint</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:simpleType>`
	`  <xs:restriction base="xs:string">`
	`  <xs:enumeration value="HARD"/>`
	`  <xs:enumeration value="SOFT"/>`
	`  </xs:restriction>`
	`  </xs:simpleType>`

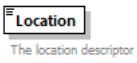
```
      </xs:element>
      <xs:element name="Location" type="xs:string">
        <xs:annotation>
          <xs:documentation>The location descriptor</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **SliceIntent/Constraints/ComponentHostingConstraints/LocationConstraint/ConstraintType**

diagram	
type	restriction of **xs:string**
properties	content simple
facets	Kind Value Annotation enumeration HARD enumeration SOFT
annotation	documentation The Type of the Constraint

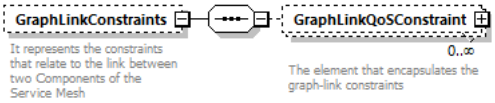
source	`<xs:element name="ConstraintType">`
	`  <xs:annotation>`
	`  <xs:documentation>The Type of the Constraint</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:simpleType>`
	`  <xs:restriction base="xs:string">`
	`  <xs:enumeration value="HARD"/>`
	`  <xs:enumeration value="SOFT"/>`
	`  </xs:restriction>`
	`  </xs:simpleType>`
	`</xs:element>`

element **SliceIntent/Constraints/ComponentHostingConstraints/LocationConstraint/Location**

diagram	
type	**xs:string**
properties	content simple
annotation	documentation The location descriptor

source	`<xs:element name="Location" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation>The location descriptor</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **SliceIntent/Constraints/GraphLinkConstraints**

diagram	
properties	minOcc 0 maxOcc 1 content complex
children	**GraphLinkQoSConstraint**
annotation	documentation It represents the constraints that relate to the link between two Components of the Service Mesh

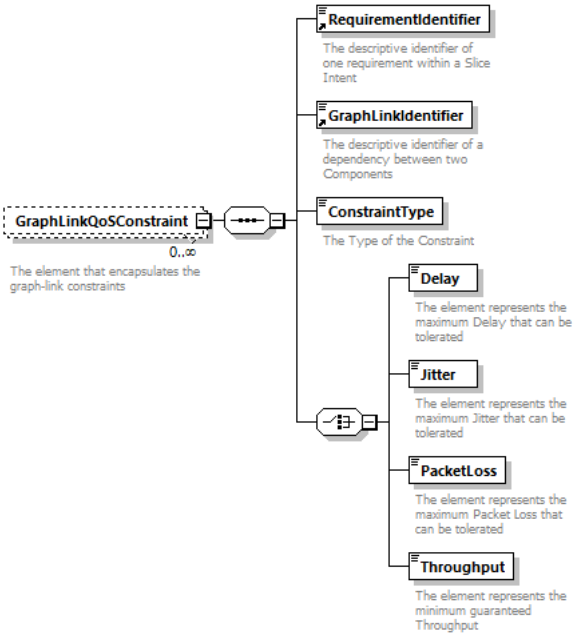
source

```
<xs:element name="GraphLinkConstraints" minOccurs="0">
  <xs:annotation>
    <xs:documentation>It represents the constraints that relate to the link
between two Components of the Service Mesh</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="GraphLinkQoSConstraint" minOccurs="0"
maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>The element that encapsulates the graph-link
constraints</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="RequirementIdentifier"/>
            <xs:element ref="GraphLinkIdentifier"/>
            <xs:element name="ConstraintType">
              <xs:annotation>
                <xs:documentation>The Type of the
Constraint</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="HARD"/>
                  <xs:enumeration value="SOFT"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:choice>
              <xs:element name="Delay" type="xs:int">
                <xs:annotation>
                  <xs:documentation>The element represents the maximum Delay
that can be tolerated</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="Jitter" type="xs:int">
                <xs:annotation>
                  <xs:documentation>The element represents the maximum Jitter
that can be tolerated</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="PacketLoss" type="xs:int">
                <xs:annotation>
                  <xs:documentation>The element represents the maximum Packet
```

```
Loss that can be tolerated</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element name="Throughput" type="xs:int">
                <xs:annotation>
                  <xs:documentation>The element represents the minimum
guaranteed Throughput</xs:documentation>
                </xs:annotation>
              </xs:element>
            </xs:choice>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **SliceIntent/Constraints/GraphLinkConstraints/GraphLinkQoSConstraint**

diagram



properties

minOcc	0
maxOcc	unbounded
content	complex

children **RequirementIdentifier GraphLinkIdentifier ConstraintType Delay Jitter PacketLoss Throughput**

annotation

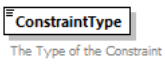
documentation
The element that encapsulates the graph-link constraints

source

```
<xs:element name="GraphLinkQoSConstraint" minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation>The element that encapsulates the graph-link
constraints</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="RequirementIdentifier"/>
      <xs:element ref="GraphLinkIdentifier"/>
```

```
      <xs:element name="ConstraintType">
        <xs:annotation>
          <xs:documentation>The Type of the Constraint</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="HARD"/>
            <xs:enumeration value="SOFT"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:choice>
        <xs:element name="Delay" type="xs:int">
          <xs:annotation>
            <xs:documentation>The element represents the maximum Delay that
can be tolerated</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Jitter" type="xs:int">
          <xs:annotation>
            <xs:documentation>The element represents the maximum Jitter that
can be tolerated</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="PacketLoss" type="xs:int">
          <xs:annotation>
            <xs:documentation>The element represents the maximum Packet Loss
that can be tolerated</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Throughput" type="xs:int">
          <xs:annotation>
            <xs:documentation>The element represents the minimum guaranteed
Throughput</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **SliceIntent/Constraints/GraphLinkConstraints/GraphLinkQoSConstraint/ConstraintType**

diagram	
type	restriction of **xs:string**
properties	content simple
facets	Kind Value Annotation enumeration HARD enumeration SOFT
annotation	documentation The Type of the Constraint
source	`<xs:element name="ConstraintType">`

```
  <xs:annotation>
    <xs:documentation>The Type of the Constraint</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="HARD"/>
      <xs:enumeration value="SOFT"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

element **SliceIntent/Constraints/GraphLinkConstraints/GraphLinkQoSConstraint/Delay**

diagram	
type	**xs:int**
properties	content simple
annotation	documentation The element represents the maximum Delay that can be tolerated

source	`<xs:element name="Delay" type="xs:int">`
	`  <xs:annotation>`
	`  <xs:documentation>The element represents the maximum Delay that can be tolerated</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **SliceIntent/Constraints/GraphLinkConstraints/GraphLinkQoSConstraint/Jitter**

diagram	
type	**xs:int**
properties	content simple
annotation	documentation The element represents the maximum Jitter that can be tolerated

source	`<xs:element name="Jitter" type="xs:int">`
	`  <xs:annotation>`
	`  <xs:documentation>The element represents the maximum Jitter that can be tolerated</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **SliceIntent/Constraints/GraphLinkConstraints/GraphLinkQoSConstraint/PacketLoss**

diagram	

type	**xs:int**
properties	content simple
annotation	documentation The element represents the maximum Packet Loss that can be tolerated

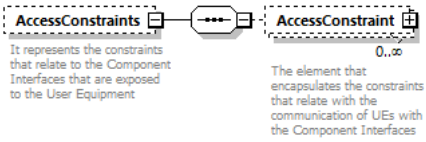
source	`<xs:element name="PacketLoss" type="xs:int">`
	`  <xs:annotation>`
	`  <xs:documentation>The element represents the maximum Packet Loss that can be tolerated</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **SliceIntent/Constraints/GraphLinkConstraints/GraphLinkQoSConstraint/Throughput**

diagram	
type	**xs:int**
properties	content simple
annotation	documentation The element represents the minimum guaranteed Throughput

source	`<xs:element name="Throughput" type="xs:int">`
	`  <xs:annotation>`
	`  <xs:documentation>The element represents the minimum guaranteed Throughput</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **SliceIntent/Constraints/AccessConstraints**

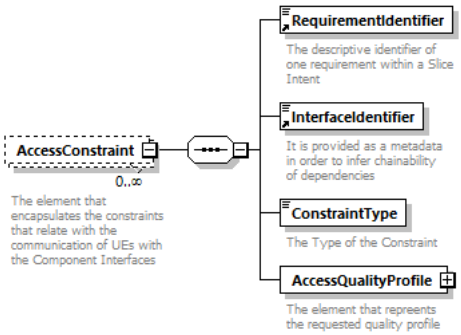
diagram	
properties	minOcc 0 maxOcc 1 content complex
children	**AccessConstraint**
annotation	documentation It represents the constraints that relate to the Component Interfaces that are exposed to the User Equipment

source	`<xs:element name="AccessConstraints" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation>It represents the constraints that relate to the Component Interfaces that are exposed to the User Equipment</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element name="AccessConstraint" minOccurs="0" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation>The element that encapsulates the constraints that`

```
relate with the communication of UEs with the Component
Interfaces</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="RequirementIdentifier"/>
            <xs:element ref="InterfaceIdentifier"/>
            <xs:element name="ConstraintType">
              <xs:annotation>
                <xs:documentation>The Type of the
Constraint</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="HARD"/>
                  <xs:enumeration value="SOFT"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="AccessQualityProfile">
              <xs:annotation>
                <xs:documentation>The element that repreents the requested
quality profile</xs:documentation>
              </xs:annotation>
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="QCI" type="xs:string">
                    <xs:annotation>
                      <xs:documentation>The QoS class
Identifier</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="ResourceType">
                    <xs:annotation>
                      <xs:documentation>The indication weather or not
Guarantted Bit Rate must be assured</xs:documentation>
                    </xs:annotation>
                    <xs:simpleType>
                      <xs:restriction base="xs:string">
                        <xs:enumeration value="GBR"/>
                        <xs:enumeration value="NONGBR"/>
                      </xs:restriction>
                    </xs:simpleType>
                  </xs:element>
                  <xs:element name="Priority" type="xs:int">
                    <xs:annotation>
                      <xs:documentation>The priority level</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="PacketDelayBudget" type="xs:int">
                    <xs:annotation>
                      <xs:documentation>The Packet Delay
Budget</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="PacketErrorLossRate">
```

```
                    <xs:annotation>
                      <xs:documentation>The Packet Error Loss
Rate</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                  <xs:element name="UEType" type="xs:string">
                    <xs:annotation>
                      <xs:documentation>The Type of User
Equipment</xs:documentation>
                    </xs:annotation>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **SliceIntent/Constraints/AccessConstraints/AccessConstraint**

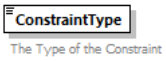
diagram	
properties	minOcc 0 maxOcc unbounded content complex
children	RequirementIdentifier InterfaceIdentifier ConstraintType AccessQualityProfile
annotation	documentation The element that encapsulates the constraints that relate with the communication of UEs with the Component Interfaces

source	`<xs:element name="AccessConstraint" minOccurs="0" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation>The element that encapsulates the constraints that relate with the communication of UEs with the Component Interfaces</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element ref="RequirementIdentifier"/>`
	`  <xs:element ref="InterfaceIdentifier"/>`
	`  <xs:element name="ConstraintType">`
	`  <xs:annotation>`
	`  <xs:documentation>The Type of the Constraint</xs:documentation>`
	`  </xs:annotation>`

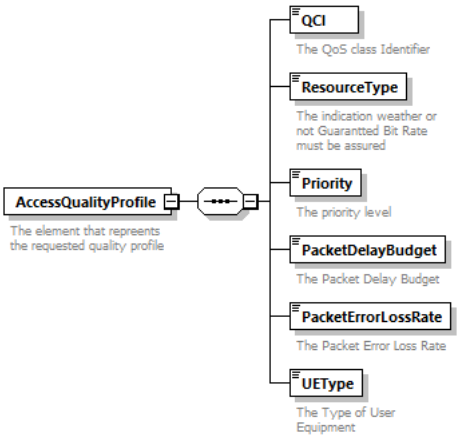
```
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="HARD"/>
            <xs:enumeration value="SOFT"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="AccessQualityProfile">
        <xs:annotation>
          <xs:documentation>The element that repreents the requested quality
profile</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="QCI" type="xs:string">
              <xs:annotation>
                <xs:documentation>The QoS class Identifier</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="ResourceType">
              <xs:annotation>
                <xs:documentation>The indication weather or not Guarantted Bit
Rate must be assured</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="GBR"/>
                  <xs:enumeration value="NONGBR"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="Priority" type="xs:int">
              <xs:annotation>
                <xs:documentation>The priority level</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="PacketDelayBudget" type="xs:int">
              <xs:annotation>
                <xs:documentation>The Packet Delay Budget</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="PacketErrorLossRate">
              <xs:annotation>
                <xs:documentation>The Packet Error Loss
Rate</xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="UEType" type="xs:string">
              <xs:annotation>
                <xs:documentation>The Type of User
Equipment</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
```

```
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **SliceIntent/Constraints/AccessConstraints/AccessConstraint/ConstraintType**

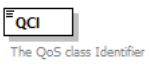
diagram	
type	restriction of **xs:string**
properties	content simple
facets	Kind Value Annotation enumeration HARD enumeration SOFT
annotation	documentation The Type of the Constraint
source	<xs:element name="ConstraintType"> <xs:annotation> <xs:documentation>The Type of the Constraint</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="HARD"/> <xs:enumeration value="SOFT"/> </xs:restriction> </xs:simpleType> </xs:element>

element **SliceIntent/Constraints/AccessConstraints/AccessConstraint/AccessQualityProfile**

diagram	
properties	content complex
children	QCI ResourceType Priority PacketDelayBudget PacketErrorLossRate UEType
annotation	documentation The element that repreents the requested quality profile
source	<xs:element name="AccessQualityProfile"> <xs:annotation> <xs:documentation>The element that repreents the requested quality

```
profile</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="QCI" type="xs:string">
        <xs:annotation>
          <xs:documentation>The QoS class Identifier</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="ResourceType">
        <xs:annotation>
          <xs:documentation>The indication weather or not Guarantted Bit Rate
must be assured</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="GBR"/>
            <xs:enumeration value="NONGBR"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Priority" type="xs:int">
        <xs:annotation>
          <xs:documentation>The priority level</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="PacketDelayBudget" type="xs:int">
        <xs:annotation>
          <xs:documentation>The Packet Delay Budget</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="PacketErrorLossRate">
        <xs:annotation>
          <xs:documentation>The Packet Error Loss Rate</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="UEType" type="xs:string">
        <xs:annotation>
          <xs:documentation>The Type of User Equipment</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

element **SliceIntent/Constraints/AccessConstraints/AccessConstraint/AccessQualityProfile/QCI**

diagram	
type	**xs:string**
properties	content simple
annotation	documentation The QoS class Identifier

source	<xs:element name="QCI" type="xs:string"> <xs:annotation> <xs:documentation>The QoS class Identifier</xs:documentation> </xs:annotation> </xs:element>

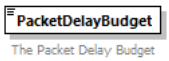
element **SliceIntent/Constraints/AccessConstraints/AccessConstraint/AccessQualityProfile/ResourceType**

diagram	ResourceType The indication weather or not Guarantted Bit Rate must be assured
type	restriction of **xs:string**
properties	content simple
facets	Kind Value Annotation enumeration GBR enumeration NONGBR
annotation	documentation The indication weather or not Guarantted Bit Rate must be assured
source	<xs:element name="ResourceType"> <xs:annotation> <xs:documentation>The indication weather or not Guarantted Bit Rate must be assured</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="GBR"/> <xs:enumeration value="NONGBR"/> </xs:restriction> </xs:simpleType> </xs:element>

element **SliceIntent/Constraints/AccessConstraints/AccessConstraint/AccessQualityProfile/Priority**

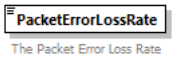
diagram	Priority The priority level
type	**xs:int**
properties	content simple
annotation	documentation The priority level
source	<xs:element name="Priority" type="xs:int"> <xs:annotation> <xs:documentation>The priority level</xs:documentation> </xs:annotation> </xs:element>

element **SliceIntent/Constraints/AccessConstraints/AccessConstraint/AccessQualityProfile/PacketDelayBudget**

diagram	
type	**xs:int**
properties	content simple
annotation	documentation The Packet Delay Budget

source	`<xs:element name="PacketDelayBudget" type="xs:int">`
	`  <xs:annotation>`
	`  <xs:documentation>The Packet Delay Budget</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **SliceIntent/Constraints/AccessConstraints/AccessConstraint/AccessQualityProfile/PacketErrorLossRate**

diagram	
annotation	documentation The Packet Error Loss Rate

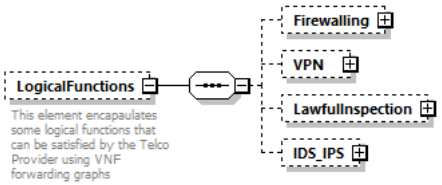
source	`<xs:element name="PacketErrorLossRate">`
	`  <xs:annotation>`
	`  <xs:documentation>The Packet Error Loss Rate</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **SliceIntent/Constraints/AccessConstraints/AccessConstraint/AccessQualityProfile/UEType**

diagram	
type	**xs:string**
properties	content simple
annotation	documentation The Type of User Equipment

source	`<xs:element name="UEType" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation>The Type of User Equipment</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

element **SliceIntent/LogicalFunctions**

diagram	
properties	minOcc 0 maxOcc 1 content complex
children	Firewalling VPN LawfulInspection IDS_IPS
annotation	documentation This element encapaulates some logical functions that can be satisfied by the Telco Provider using VNF forwarding graphs
source	<xs:element name="LogicalFunctions" minOccurs="0"> <xs:annotation> <xs:documentation>This element encapaulates some logical functions that can be satisfied by the Telco Provider using VNF forwarding graphs</xs:documentation> </xs:annotation> <xs:complexType> <xs:sequence> <xs:element name="Firewalling" minOccurs="0"> <xs:complexType> <xs:sequence> <xs:element name="FirewallConfiguration" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="VPN" minOccurs="0"> <xs:complexType> <xs:sequence> <xs:element name="VPNConfiguration" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="LawfulInspection" minOccurs="0"> <xs:complexType> <xs:sequence> <xs:element name="LawfulInspectionConfiguration" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> <xs:element name="IDS_IPS" minOccurs="0"> <xs:complexType> <xs:sequence> <xs:element name="IDSIPSConfiguration" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </xs:element> </xs:sequence>

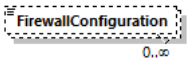
```
  </xs:complexType>
</xs:element>
```

element **SliceIntent/LogicalFunctions/Firewalling**

diagram	
properties	minOcc 0 maxOcc 1 content complex
children	**FirewallConfiguration**

source	`<xs:element name="Firewalling" minOccurs="0">`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element name="FirewallConfiguration" minOccurs="0" maxOccurs="unbounded"/>`
	`  </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

element **SliceIntent/LogicalFunctions/Firewalling/FirewallConfiguration**

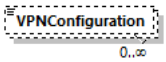
diagram	
properties	minOcc 0 maxOcc unbounded
source	`<xs:element name="FirewallConfiguration" minOccurs="0" maxOccurs="unbounded"/>`

element **SliceIntent/LogicalFunctions/VPN**

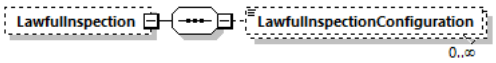
diagram	
properties	minOcc 0 maxOcc 1 content complex
children	**VPNConfiguration**

source	`<xs:element name="VPN" minOccurs="0">`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element name="VPNConfiguration" minOccurs="0" maxOccurs="unbounded"/>`
	`  </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

element **SliceIntent/LogicalFunctions/VPN/VPNConfiguration**

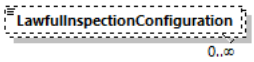
diagram	
properties	minOcc 0 maxOcc unbounded
source	`<xs:element name="VPNConfiguration" minOccurs="0" maxOccurs="unbounded"/>`

element **SliceIntent/LogicalFunctions/LawfulInspection**

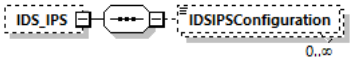
diagram	
properties	minOcc 0 maxOcc 1 content complex
children	**LawfulInspectionConfiguration**

source	`<xs:element name="LawfulInspection" minOccurs="0">`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element name="LawfulInspectionConfiguration" minOccurs="0" maxOccurs="unbounded"/>`
	`  </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

element **SliceIntent/LogicalFunctions/LawfulInspection/LawfulInspectionConfiguration**

diagram	
properties	minOcc 0 maxOcc unbounded
source	`<xs:element name="LawfulInspectionConfiguration" minOccurs="0" maxOccurs="unbounded"/>`

element **SliceIntent/LogicalFunctions/IDS_IPS**

diagram	
properties	minOcc 0 maxOcc 1 content complex
children	**IDSIPSConfiguration**

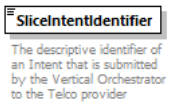
source	`<xs:element name="IDS_IPS" minOccurs="0">`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element name="IDSIPSConfiguration" minOccurs="0" maxOccurs="unbounded"/>`
	`  </xs:sequence>`
	`  </xs:complexType>`

	</xs:element>

element **SliceIntent/LogicalFunctions/IDS_IPS/IDSIPSConfiguration**

diagram	
properties	minOcc 0 maxOcc unbounded
source	<xs:element name="IDSIPSConfiguration" minOccurs="0" maxOccurs="unbounded"/>

element **SliceIntentIdentifier**

diagram	
type	**xs:string**
properties	content simple
used by	elements **Slice SliceIntent**
annotation	documentation The descriptive identifier of an Intent that is submitted by the Vertical Orchestrator to the Telco provider
source	<xs:element name="SliceIntentIdentifier" type="xs:string"> <xs:annotation> <xs:documentation>The descriptive identifier of an Intent that is submitted by the Vertical Orchestrator to the Telco provider</xs:documentation> </xs:annotation> </xs:element>

element **VIMIdentifier**

diagram	
type	**xs:string**
properties	content simple
used by	elements **Slice/ComponentPlacementDescriptor/ComponentPlacement Slice/VIMDescriptor/VIM**
annotation	documentation The descriptive identifier of a VIM as advertised by a Telco provider
source	<xs:element name="VIMIdentifier" type="xs:string"> <xs:annotation> <xs:documentation>The descriptive identifier of a VIM as advertised by a Telco provider</xs:documentation> </xs:annotation> </xs:element>