Schema documentation for DR-GW-Call.Events.xsd

november 5, 2024

Table of Contents

Namespace: "DR-GW-Interface/DR-GW-Call.Events" 2
Schema(s) 2
Main schema DR-GW-Call.Events.xsd 2
Element(s) 3
Element Call_Response 3
Element Call_SelectEvent 3
Element Call_SelectEvent / sel 4
Element Call_Event 4
Element Call_Event / tetraCallId 6
Element Call_Event / action 6
Element Call_Event / attributes 6
Element Call_Event / callingParty 7
Element Call_Event / calledParty 8
Element Call_Event / disconnectCause 9
Element Call_PTTEvent 10
Element Call_PTTEvent / tetraCallId 11
Element Call_PTTEvent / granted 11
Element Call_PTTEvent / ceased 12
Element Call_PTTEvent / wait 12
Element Call_UnitInEmergencyEvent 12
Element Call_UnitInEmergencyEvent / group 14
Element Call_UnitInEmergencyEvent / tetraCallId 14
Element Call_UnitInEmergencyEvent / unitInEmg 14
Element Call_UnitInEmergencyEvent / unitInEmgType 15
Element Call_UnitInEmergencyEvent / emgInfo 15
Element Call_UnitInEmergencyEvent / tstamp 16
Element Call_KeyExchangeEvent 16
Element Call_KeyExchangeEvent / state 17
Element Call_KeyExchangeEvent / code 18
Element Call_KeyExchangeEvent / priority 18
Element Call_KeyExchangeEvent / interaction 18
Element Call_KeyExchangeEvent / text 19
Element Call_KeyExchangeEvent / tone 19
Namespace: "DR-GW-Interface/CommonTypes" 19
Schema(s) 19
Imported schema CommonTypes.xsd 19
Element(s) 19
Element ct:typeResponse / ct:requestId 19
Element ct:typeResponse / ct:result 20
Element ct:typeResult / ct:responseCode 20
Element ct:typeResult / ct:sourceSystem 20
Element ct:typeResult / ct:result 21
Element ct:typeEvent / ct:requestId 21
Element ct:typeEvent / ct:result 21
Element ct:typeAddress / ct:subscriber 21
Element ct:typeSubscriberAddress / ct:ssi 22
Element ct:typeSubscriberAddress / ct:tsi 22
Element ct:typeTSI / ct:mnc 23
Element ct:typeTSI / ct:mcc 23
Element ct:typeTSI / ct:ssi 23
Element ct:typeAddress / ct:alias 23
Element ct:typeAddress / ct:msisdn 23
Element ct:typeAddress / ct:fssn 24
Element ct:typeAddress / ct:external 24
Element ct:typeExternal / ct:gatewayNumber 24
Element ct:typeExternal / ct:number 25
Element ct:typeAddress / ct:opta 25
Element ct:typeAddress / ct:cell 25
Element ct:typeRequest / ct:requestId 25
Complex Type(s) 26

Complex Type ct:typeResponse ... 26
Complex Type ct:typeResult ... 26
Complex Type ct:typeEvent ... 26
Complex Type ct:typeAddress ... 27
Complex Type ct:typeSubscriberAddress ... 27
Complex Type ct:typeTSI ... 28
Complex Type ct:typeExternal ... 28
Complex Type ct:typeEmpty ... 29
Complex Type ct:typeRequest ... 29
Simple Type(s) ... 29
Simple Type ct:typeResponseCode ... 29
Simple Type ct:typeSourceSystem ... 30
Simple Type ct:typeDialString ... 30
Simple Type ct:typeOPTA ... 30
Simple Type ct:typeAddressingStyle ... 31
Namespace: "DR-GW-Interface/DR-GW-Call.CommonTypes" ... 31
Schema(s) ... 31
Imported schema DR-GW-Call.CommonTypes.xsd ... 31
Element(s) ... 31
Element typeSelection / level ... 31
Element typeSelection / target ... 32
Element typeCallAttributes / hook ... 32
Element typeCallAttributes / mode ... 32
Element typeCallAttributes / commtype ... 33
Element typeCallAttributes / priority ... 33
Element typeCallAttributes / encryption ... 33
Element typeCallAttributes / ambienceListen ... 33
Element typeCallAttributes / req2speak ... 34
Element typeCallAttributes / demandPriority ... 34
Element typeDisconnectCause / protocol ... 34
Element typeDisconnectCause / code ... 35
Element typeDisconnectCause / text ... 35
Element typeTxGranted / txGrant ... 35
Element typeTxGranted / talkingParty ... 36
Element typeTxGranted / encryption ... 36
Element typeTxGranted / txPriority ... 37
Element typeTxGranted / txInterrupt ... 37
Element typeTxGranted / txRepeat ... 37
Element typeTxGranted / workstationId ... 38
Complex Type(s) ... 38
Complex Type typeSelection ... 38
Complex Type typeCallAttributes ... 38
Complex Type typeDisconnectCause ... 39
Complex Type typeTxGranted ... 40
Simple Type(s) ... 41
Simple Type typeSelectionLevel ... 41
Simple Type typeActionEvent ... 42
Simple Type typeCallMode ... 43
Simple Type typeCallType ... 43
Simple Type typeTxDemandPriority ... 43
Simple Type typeTxGrant ... 44
Simple Type typeTxPriority ... 44
Simple Type typeUnitInEmergencyType ... 45
Simple Type typeEmergencyInfo ... 45
Simple Type typeKeyExchangeState ... 46
Simple Type typeKeyExchangeCode ... 46
Simple Type typeKeyExchangeTextPriority ... 47
Simple Type typeKeyExchangeText ... 47
Simple Type typeActionPTTRequest ... 47
Simple Type typeActionRequest ... 48
Simple Type typeAudioCodec ... 49
Simple Type typeWorkstationId ... 49
Simple Type typeKeyExchangeAction ... 50
Simple Type typeKeyManagementTextPriority ... 50

Namespace: "DR-GW-Interface/DR-GW-Call.Events"

Schema(s)

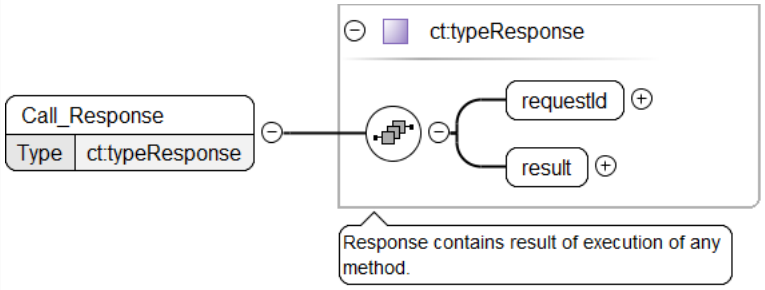
Main schema DR-GW-Call.Events.xsd

Namespace	DR-GW-Interface/DR-GW-Call.Events

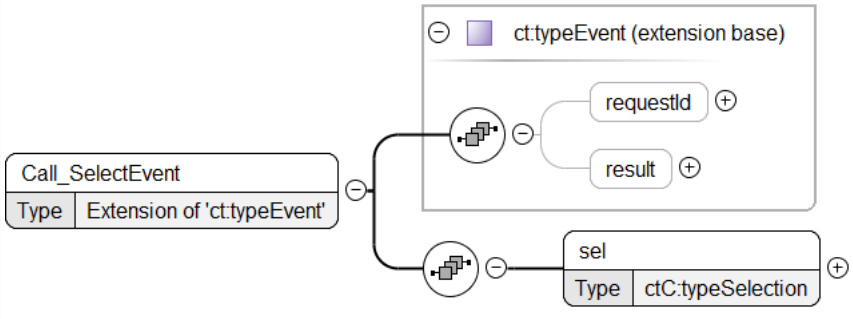
Annotations	Version 1.1.1
Properties	attribute form default: unqualified element form default: qualified

Element(s)

Element Call_Response

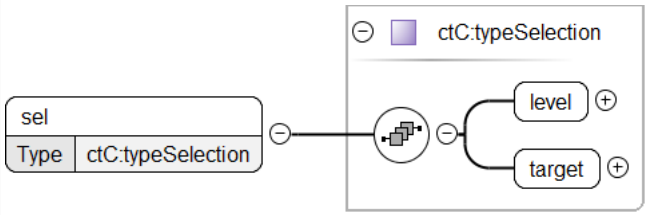
Namespace	DR-GW-Interface/DR-GW-Call.Events
Annotations	
Diagram	
Type	ct:typeResponse
Properties	content: complex
Model	ct:requestId , ct:result
Children	ct:requestId, ct:result
Instance	<Call_Response xmlns="DR-GW-Interface/DR-GW-Call.Events" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:requestId>{1,1}</ct:requestId> <ct:result>{1,1}</ct:result> </Call_Response>
Source	<xs:element name="Call_Response" type="ct:typeResponse"> <xs:annotation> <xs:documentation/> </xs:annotation> </xs:element>

Element Call_SelectEvent

Namespace	DR-GW-Interface/DR-GW-Call.Events
Annotations	
Diagram	
Type	extension of ct:typeEvent
Type hierarchy	• ct:typeEvent
Properties	content: complex
Model	ct:requestId{0,1} , ct:result{0,1} , sel
Children	ct:requestId, ct:result, sel
Instance	<Call_SelectEvent xmlns="DR-GW-Interface/DR-GW-Call.Events" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:requestId>{0,1}</ct:requestId> <ct:result>{0,1}</ct:result>

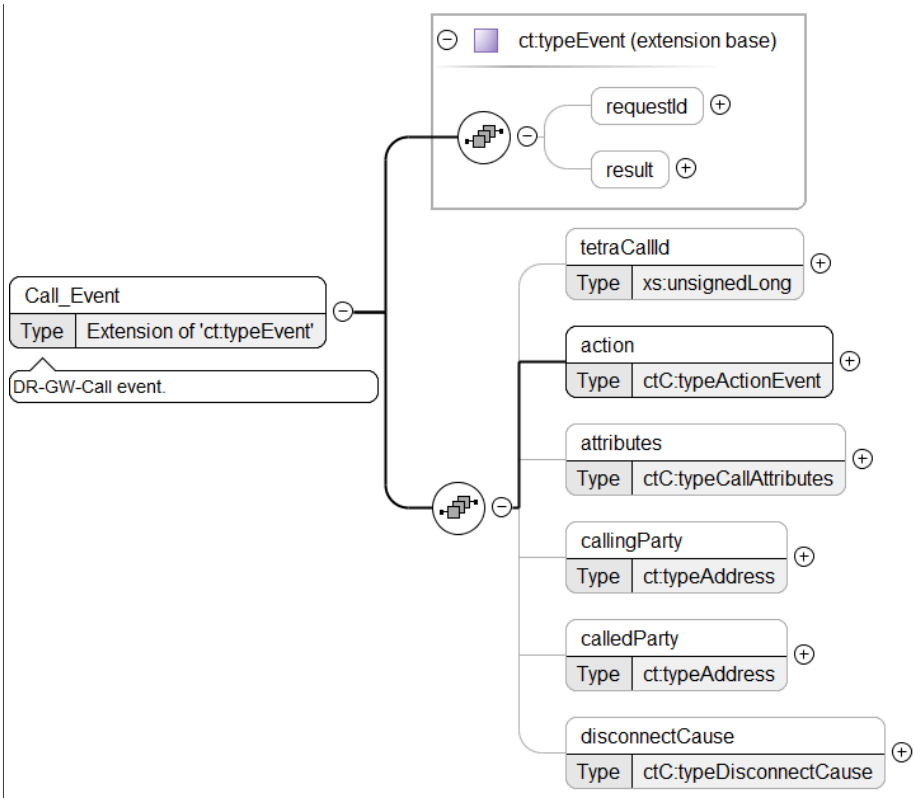
	<sel>{1,1}</sel> </Call_SelectEvent>
Source	<xs:element name="Call_SelectEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexType> <xs:complexContent> <xs:extension base="ct:typeEvent"> <xs:sequence> <xs:element name="sel" type="ctC:typeSelection"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </xs:element>

Element Call_SelectEvent / sel

Namespace	DR-GW-Interface/DR-GW-Call.Events
Diagram	
Type	typeSelection
Properties	content: complex
Model	level , target
Children	level, target
Instance	<sel xmlns="DR-GW-Interface/DR-GW-Call.Events" xmlns:ctC="DR-GW-Interface/DR-GW-Call.CommonTypes"> <ctC:level>{1,1}</ctC:level> <ctC:target>{1,1}</ctC:target> </sel>
Source	<xs:element name="sel" type="ctC:typeSelection"/>

Element Call_Event

Namespace	DR-GW-Interface/DR-GW-Call.Events
Annotations	DR-GW-Call event.

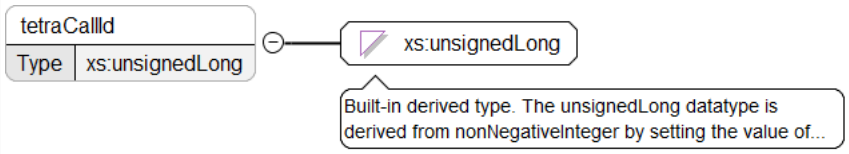
Diagram	
Type	extension of ct:typeEvent
Type hierarchy	• ct:typeEvent
Properties	content: complex
Model	ct:requestId{0,1} , ct:result{0,1} , tetraCallId{0,1} , action , attributes{0,1} , callingParty{0,1} , calledParty{0,1} , disconnectCause{0,1}
Children	action, attributes, calledParty, callingParty, ct:requestId, ct:result, disconnectCause, tetraCallId

Instance	<pre><Call_Event xmlns="DR-GW-Interface/DR-GW-Call.Events" xmlns:ct="DR-GW-Interface/CommonTypes">
	  <ct:requestId>{0,1}</ct:requestId>
	  <ct:result>{0,1}</ct:result>
	  <tetraCallId>{0,1}</tetraCallId>
	  <action>{1,1}</action>
	  <attributes>{0,1}</attributes>
	  <callingParty>{0,1}</callingParty>
	  <calledParty>{0,1}</calledParty>
	  <disconnectCause>{0,1}</disconnectCause>
	</Call_Event></pre>

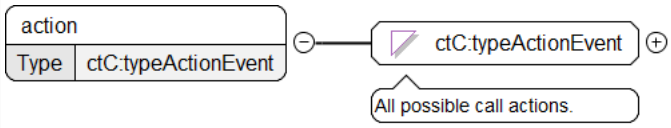
Source	<pre><xs:element name="Call_Event">
	  <xs:annotation>
	  <xs:documentation>DR-GW-Call event.</xs:documentation>
	  </xs:annotation>
	  <xs:complexType>
	  <xs:complexContent>
	  <xs:extension base="ct:typeEvent">
	  <xs:sequence>
	  <xs:element name="tetraCallId" type="xs:unsignedLong" minOccurs="0"/>
	  <xs:element name="action" type="ctC:typeActionEvent"/>
	  <xs:element name="attributes" type="ctC:typeCallAttributes" minOccurs="0"/>
	  <xs:element name="callingParty" type="ct:typeAddress" minOccurs="0"/>
	  <xs:element name="calledParty" type="ct:typeAddress" minOccurs="0"/>
	  <!--
	removed as CR-36 replaces it
	  <xs:element name="keymngstate" type="ctC:typeKeyManagement" minOccurs="0">
	  <xs:annotation>
	  <xs:documentation>If present, it informs about
	  the current state of keymanagement.</xs:documentation>
	  </xs:annotation>
	  </xs:element>
	 -->
	  <xs:element name="disconnectCause" type="ctC:typeDisconnectCause" minOccurs="0"/>
	  </xs:sequence>
	  </xs:extension></pre>

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

Element Call_Event / tetraCallId

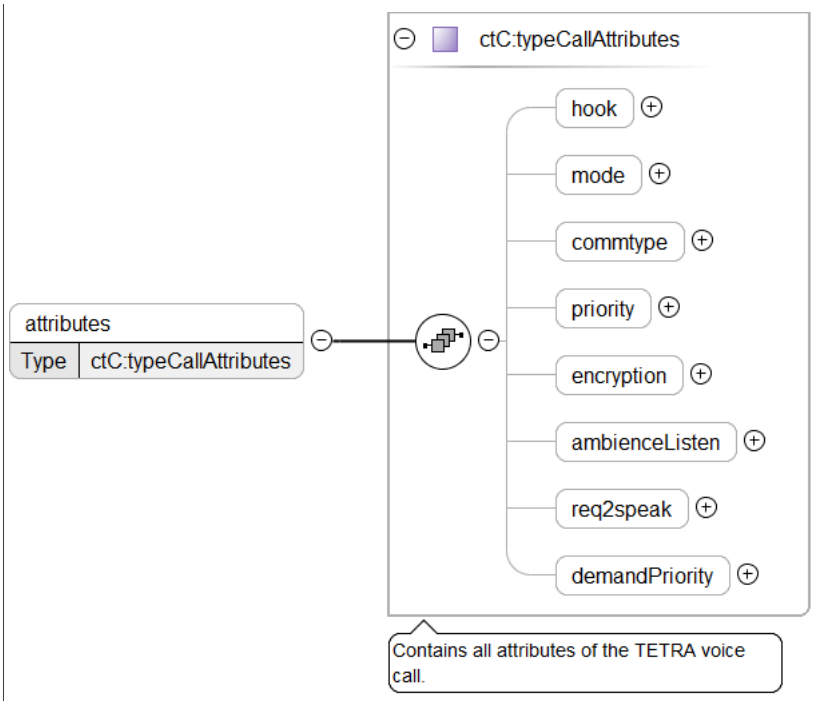
Namespace	DR-GW-Interface/DR-GW-Call.Events
Diagram	
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	`<xs:element name="tetraCallId" type="xs:unsignedLong" minOccurs="0"/>`

Element Call_Event / action

Namespace	DR-GW-Interface/DR-GW-Call.Events
Diagram	
Type	typeActionEvent
Properties	content: simple
Facets	enumeration: incoming — This event fired when there is an incoming call. This is the first indication of a new incoming call. enumeration: connected — This event is used to inform that call has been connected and call setup is finished. enumeration: held — This event is used to inform TCS Client that individual call was put to hold. enumeration: resumed — This event is used to inform that individual call has been taken from hold. enumeration: disconnected — This event is used to inform that the call was disconnected. enumeration: transferred — This event is a response to transfer method call and indicates the result of the request.
Source	`<xs:element name="action" type="ctC:typeActionEvent"/>`

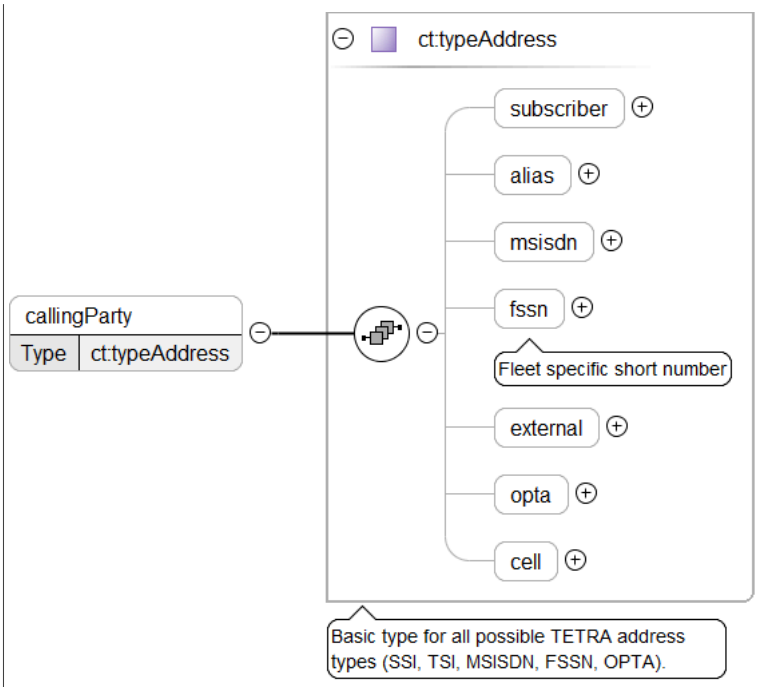
Element Call_Event / attributes

Namespace	DR-GW-Interface/DR-GW-Call.Events

Diagram	
Type	typeCallAttributes
Properties	content: complex minOccurs: 0
Model	hook{0,1} , mode{0,1} , commtype{0,1} , priority{0,1} , encryption{0,1} , ambienceListen{0,1} , req2speak{0,1} , demandPriority{0,1}
Children	ambienceListen, commtype, demandPriority, encryption, hook, mode, priority, req2speak
Instance	<attributes xmlns="DR-GW-Interface/DR-GW-Call.Events" xmlns:ctC="DR-GW-Interface/DR-GW-Call.CommonTypes"> <ctC:hook>{0,1}</ctC:hook> <ctC:mode>{0,1}</ctC:mode> <ctC:commtype>{0,1}</ctC:commtype> <ctC:priority>{0,1}</ctC:priority> <ctC:encryption>{0,1}</ctC:encryption> <ctC:ambienceListen>{0,1}</ctC:ambienceListen> <ctC:req2speak>{0,1}</ctC:req2speak> <ctC:demandPriority>{0,1}</ctC:demandPriority> </attributes>
Source	<xs:element name="attributes" type="ctC:typeCallAttributes" minOccurs="0"/>

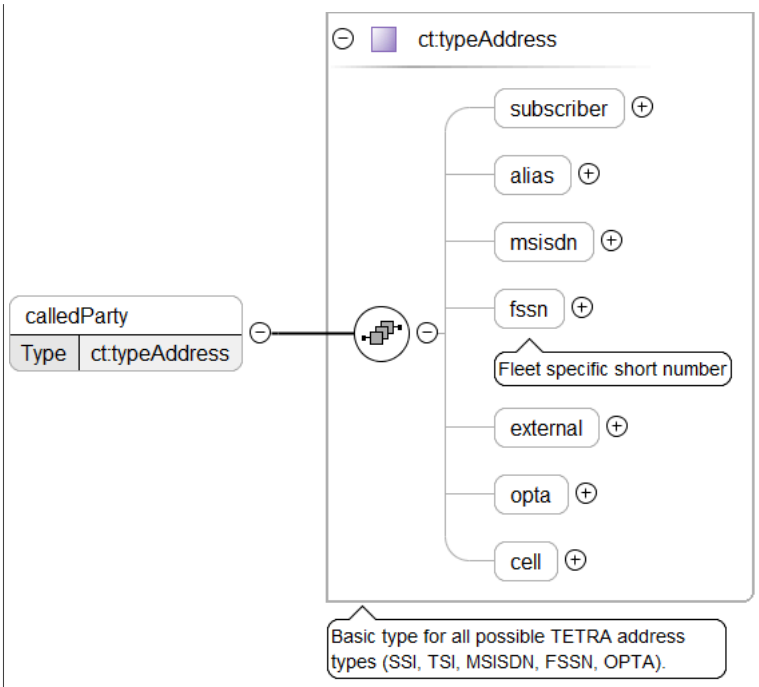
Element Call_Event / callingParty

Namespace	DR-GW-Interface/DR-GW-Call.Events

Diagram	
Type	ct:typeAddress
Properties	content: complex minOccurs: 0
Model	ct:subscriber{0,1} , ct:alias{0,1} , ct:msisdn{0,1} , ct:fssn{0,1} , ct:external{0,1} , ct:opta{0,1} , ct:cell{0,1}
Children	ct:alias, ct:cell, ct:external, ct:fssn, ct:msisdn, ct:opta, ct:subscriber
Instance	<callingParty xmlns="DR-GW-Interface/DR-GW-Call.Events" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:subscriber>{0,1}</ct:subscriber> <ct:alias>{0,1}</ct:alias> <ct:msisdn>{0,1}</ct:msisdn> <ct:fssn>{0,1}</ct:fssn> <ct:external>{0,1}</ct:external> <ct:opta>{0,1}</ct:opta> <ct:cell>{0,1}</ct:cell> </callingParty>
Source	<xs:element name="callingParty" type="ct:typeAddress" minOccurs="0"/>

Element Call_Event / calledParty

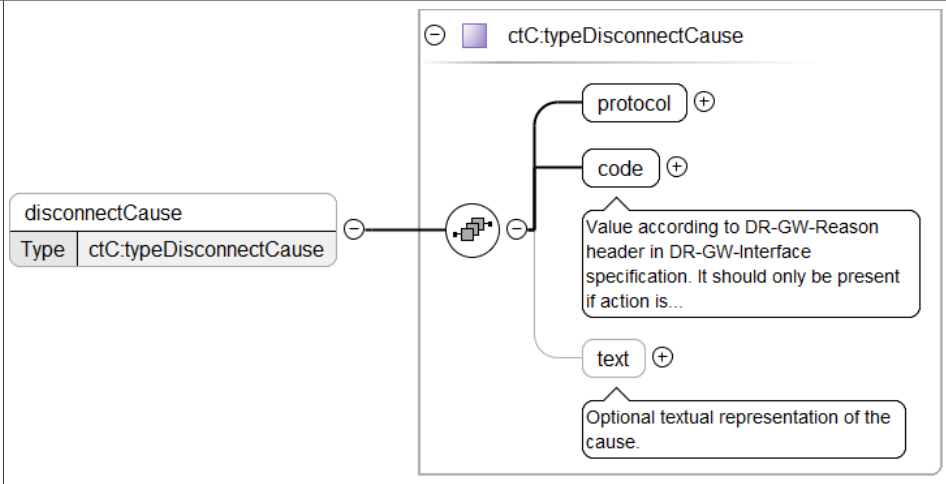
Namespace	DR-GW-Interface/DR-GW-Call.Events

Diagram	
Type	ct:typeAddress
Properties	content: complex minOccurs: 0
Model	ct:subscriber{0,1} , ct:alias{0,1} , ct:msisdn{0,1} , ct:fssn{0,1} , ct:external{0,1} , ct:opta{0,1} , ct:cell{0,1}
Children	ct:alias, ct:cell, ct:external, ct:fssn, ct:msisdn, ct:opta, ct:subscriber

Instance	<pre><calledParty xmlns="DR-GW-Interface/DR-GW-Call.Events" xmlns:ct="DR-GW-Interface/CommonTypes">
	  <ct:subscriber>{0,1}</ct:subscriber>
	  <ct:alias>{0,1}</ct:alias>
	  <ct:msisdn>{0,1}</ct:msisdn>
	  <ct:fssn>{0,1}</ct:fssn>
	  <ct:external>{0,1}</ct:external>
	  <ct:opta>{0,1}</ct:opta>
	  <ct:cell>{0,1}</ct:cell>
	</calledParty></pre>

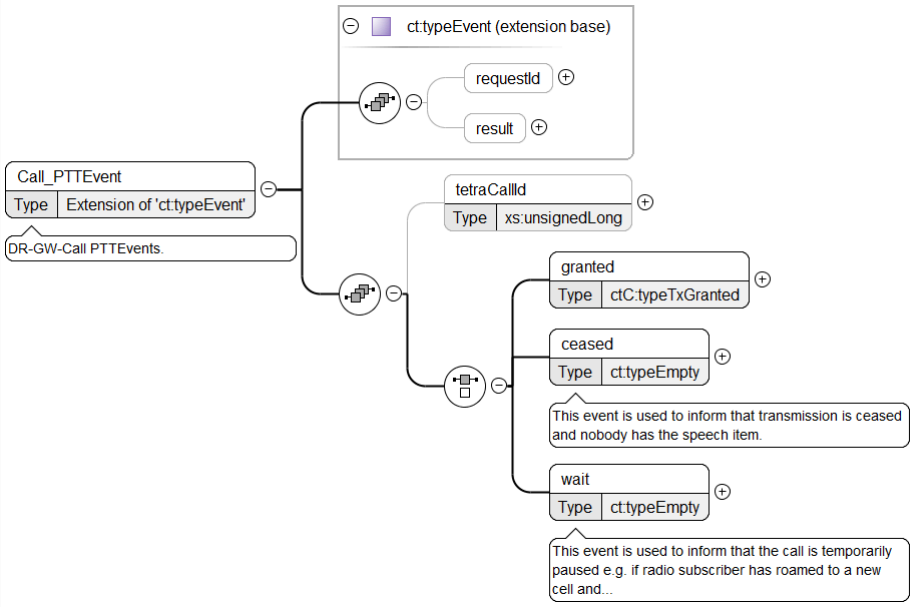
Source	<pre><xs:element name="calledParty" type="ct:typeAddress" minOccurs="0"/></pre>

Element `Call_Event / disconnectCause`

Namespace	DR-GW-Interface/DR-GW-Call.Events
Diagram	
Type	typeDisconnectCause
Properties	content: complex minOccurs: 0

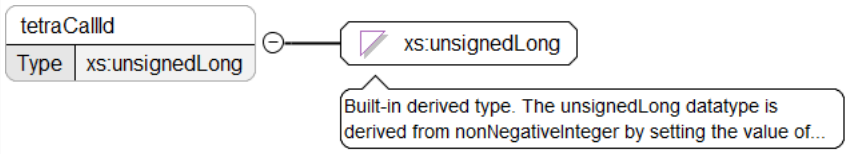
Model	protocol , code , text{0,1}
Children	code, protocol, text
Instance	<disconnectCause xmlns="DR-GW-Interface/DR-GW-Call.Events" xmlns:ctC="DR-GW-Interface/DR-GW-Call.CommonTypes"> <ctC:protocol>{1,1}</ctC:protocol> <ctC:code>{1,1}</ctC:code> <ctC:text>{0,1}</ctC:text> </disconnectCause>
Source	<xs:element name="disconnectCause" type="ctC:typeDisconnectCause" minOccurs="0"/>

Element `Call_PTTEvent`

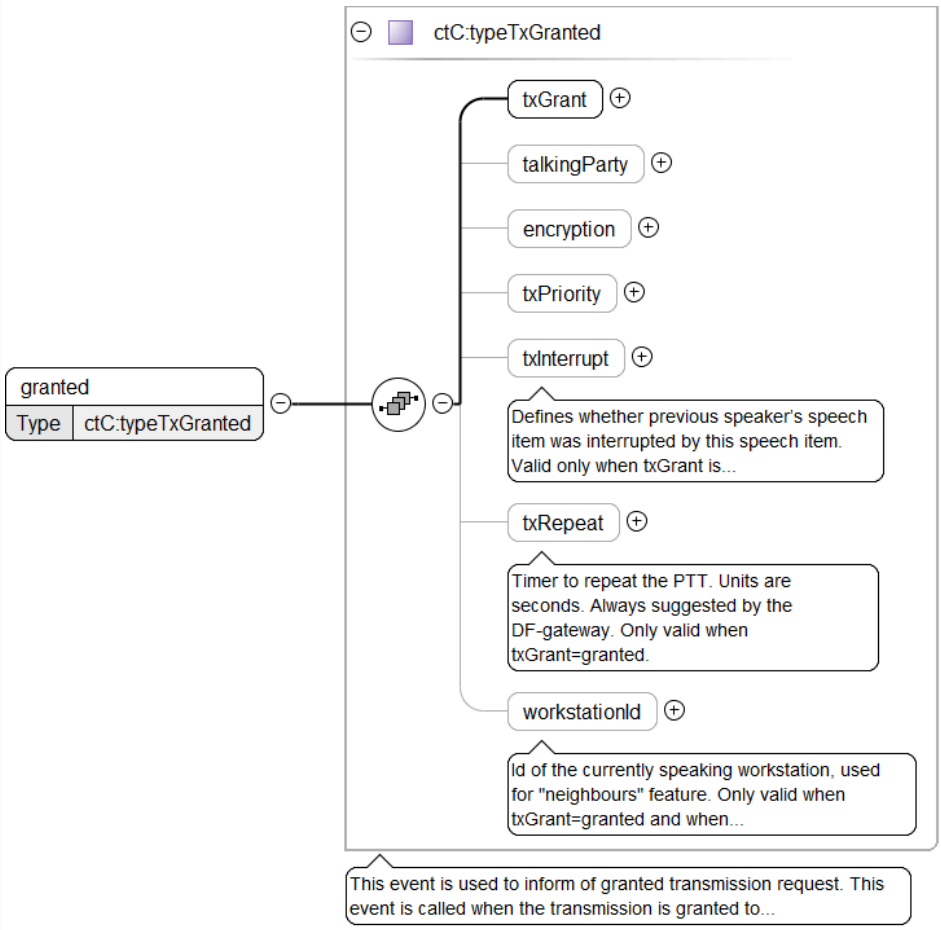
Namespace	DR-GW-Interface/DR-GW-Call.Events
Annotations	DR-GW-Call PTTEvents.
Diagram	
Type	extension of ct:typeEvent
Type hierarchy	• ct:typeEvent
Properties	content: complex
Model	ct:requestId{0,1} , ct:result{0,1} , tetraCallId{0,1} , (granted | ceased | wait)
Children	ceased, ct:requestId, ct:result, granted, tetraCallId, wait
Instance	<Call_PTTEvent xmlns="DR-GW-Interface/DR-GW-Call.Events" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:requestId>{0,1}</ct:requestId> <ct:result>{0,1}</ct:result> <tetraCallId>{0,1}</tetraCallId> <granted>{1,1}</granted> <ceased>{1,1}</ceased> <wait>{1,1}</wait> </Call_PTTEvent>
Source	<xs:element name="Call_PTTEvent"> <xs:annotation> <xs:documentation>DR-GW-Call PTTEvents.</xs:documentation> </xs:annotation> <xs:complexType> <xs:complexContent> <xs:extension base="ct:typeEvent"> <xs:sequence> <xs:element name="tetraCallId" type="xs:unsignedLong" minOccurs="0"/> <xs:choice> <xs:element name="granted" type="ctC:typeTxGranted"/> <xs:element name="ceased" type="ct:typeEmpty"> <xs:annotation> <xs:documentation>This event is used to inform that transmission is ceased and nobody has the speech item.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="wait" type="ct:typeEmpty">

```
            <xs:annotation>
              <xs:documentation>This event is used to inform that the call is temporarily paused
 e.g. if radio subscriber has roamed to a new cell and there are no free resources available.</
xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:choice>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
 </xs:complexType>
</xs:element>
```

Element `Call_PTTEvent / tetraCallId`

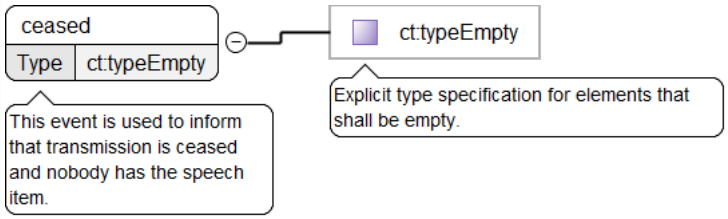
Namespace	DR-GW-Interface/DR-GW-Call.Events
Diagram	
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	`<xs:element name="tetraCallId" type="xs:unsignedLong" minOccurs="0"/>`

Element `Call_PTTEvent / granted`

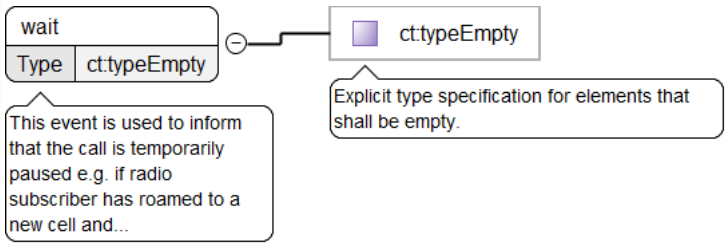
Namespace	DR-GW-Interface/DR-GW-Call.Events
Diagram	
Type	typeTxGranted
Properties	content: complex

Model	txGrant , talkingParty{0,1} , encryption{0,1} , txPriority{0,1} , txInterrupt{0,1} , txRepeat{0,1} , workstationId{0,1}
Children	encryption, talkingParty, txGrant, txInterrupt, txPriority, txRepeat, workstationId
Instance	<granted xmlns="DR-GW-Interface/DR-GW-Call.Events" xmlns:ctC="DR-GW-Interface/DR-GW-Call.CommonTypes"> <ctC:txGrant>{1,1}</ctC:txGrant> <ctC:talkingParty>{0,1}</ctC:talkingParty> <ctC:encryption>{0,1}</ctC:encryption> <ctC:txPriority>{0,1}</ctC:txPriority> <ctC:txInterrupt>{0,1}</ctC:txInterrupt> <ctC:txRepeat>{0,1}</ctC:txRepeat> <ctC:workstationId>{0,1}</ctC:workstationId> </granted>
Source	<xs:element name="granted" type="ctC:typeTxGranted"/>

Element Call_PTTEvent / ceased

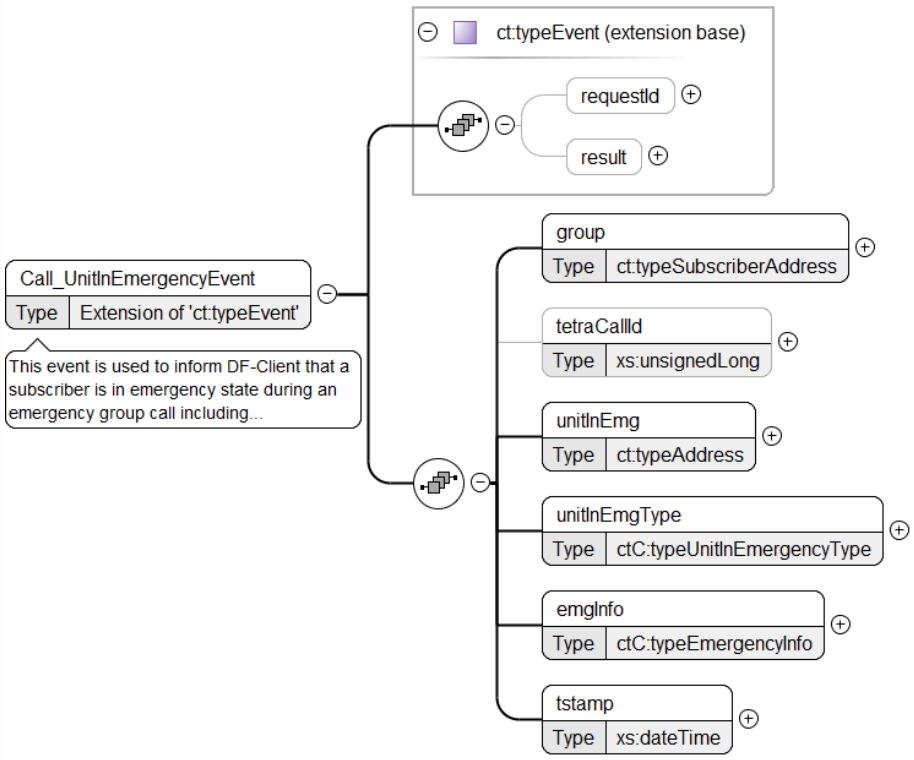
Namespace	DR-GW-Interface/DR-GW-Call.Events
Annotations	This event is used to inform that transmission is ceased and nobody has the speech item.
Diagram	
Type	ct:typeEmpty
Properties	content: complex
Source	<xs:element name="ceased" type="ct:typeEmpty"> <xs:annotation> <xs:documentation>This event is used to inform that transmission is ceased and nobody has the speech item.</xs:documentation> </xs:annotation> </xs:element>

Element Call_PTTEvent / wait

Namespace	DR-GW-Interface/DR-GW-Call.Events
Annotations	This event is used to inform that the call is temporarily paused e.g. if radio subscriber has roamed to a new cell and there are no free resources available.
Diagram	
Type	ct:typeEmpty
Properties	content: complex
Source	<xs:element name="wait" type="ct:typeEmpty"> <xs:annotation> <xs:documentation>This event is used to inform that the call is temporarily paused e.g. if radio subscriber has roamed to a new cell and there are no free resources available.</xs:documentation> </xs:annotation> </xs:element>

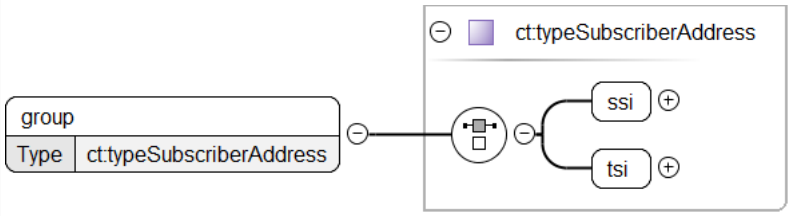
Element Call_UnitInEmergencyEvent

Namespace	DR-GW-Interface/DR-GW-Call.Events
Annotations	This event is used to inform DF-Client that a subscriber is in emergency state during an emergency group call including the ending of its emergency situation. Also queuing of emergency speech item requests is indicated using this event. Event is fired every time the TETRA system informs the Gateway that subscriber's emergency information is changed. For example,

	based on this information TCS Client could use pre-emptive speech item to request the current speaker to stop in order to let the unit in emergency to speak.
Diagram	
Type	extension of ct:typeEvent
Type hierarchy	• ct:typeEvent
Properties	content: complex
Model	ct:requestId{0,1} , ct:result{0,1} , group , tetraCallId{0,1} , unitInEmg , unitInEmgType , emgInfo , tstamp
Children	ct:requestId, ct:result, emgInfo, group, tetraCallId, tstamp, unitInEmg, unitInEmgType
Instance	<Call_UnitInEmergencyEvent xmlns="DR-GW-Interface/DR-GW-Call.Events" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:requestId>{0,1}</ct:requestId> <ct:result>{0,1}</ct:result> <group>{1,1}</group> <tetraCallId>{0,1}</tetraCallId> <unitInEmg>{1,1}</unitInEmg> <unitInEmgType>{1,1}</unitInEmgType> <emgInfo>{1,1}</emgInfo> <tstamp>{1,1}</tstamp> </Call_UnitInEmergencyEvent>
Source	<xs:element name="Call_UnitInEmergencyEvent"> <xs:annotation> <xs:documentation>This event is used to inform DF-Client that a subscriber is in emergency state during an emergency group call including the ending of its emergency situation. Also queuing of emergency speech item requests is indicated using this event. Event is fired every time the TETRA system informs the Gateway that subscriber's emergency information is changed. For example, based on this information TCS Client could use pre-emptive speech item to request the current speaker to stop in order to let the unit in emergency to speak.</xs:documentation> </xs:annotation> <xs:complexType> <xs:complexContent> <xs:extension base="ct:typeEvent"> <xs:sequence> <xs:element name="group" type="ct:typeSubscriberAddress"/> <xs:element name="tetraCallId" type="xs:unsignedLong" minOccurs="0"/> <xs:element name="unitInEmg" type="ct:typeAddress"/> <xs:element name="unitInEmgType" type="ctC:typeUnitInEmergencyType"/> <xs:element name="emgInfo" type="ctC:typeEmergencyInfo"/> <xs:element name="tstamp" type="xs:dateTime"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

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

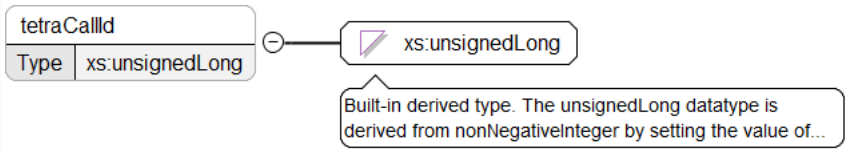
Element Call_UnitInEmergencyEvent / group

Namespace	DR-GW-Interface/DR-GW-Call.Events
Diagram	
Type	ct:typeSubscriberAddress
Properties	content: complex
Model	ct:ssi | ct:tsi
Children	ct:ssi, ct:tsi

Instance	`<group xmlns="DR-GW-Interface/DR-GW-Call.Events" xmlns:ct="DR-GW-Interface/CommonTypes">`
	`  <ct:ssi>{1,1}</ct:ssi>`
	`  <ct:tsi>{1,1}</ct:tsi>`
	`</group>`

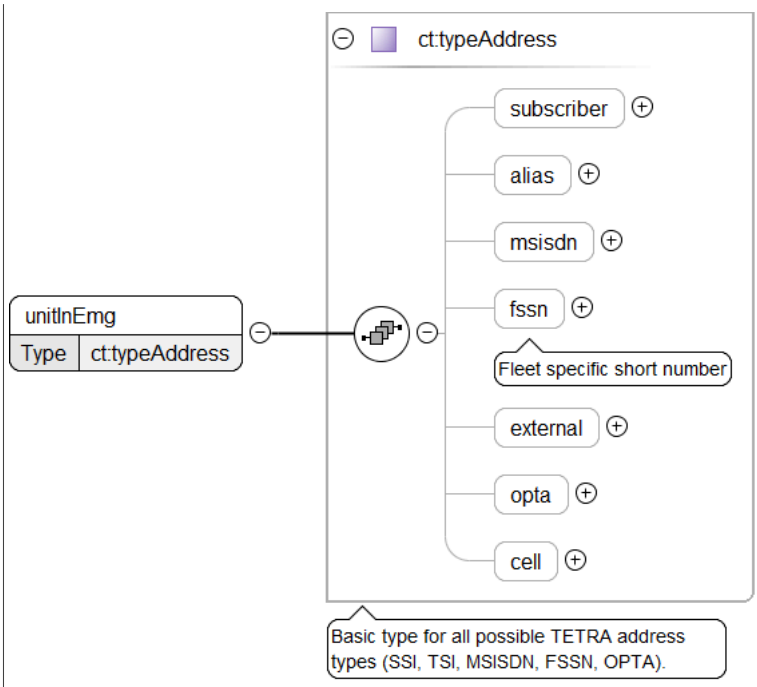
Source	`<xs:element name="group" type="ct:typeSubscriberAddress"/>`

Element Call_UnitInEmergencyEvent / tetraCallId

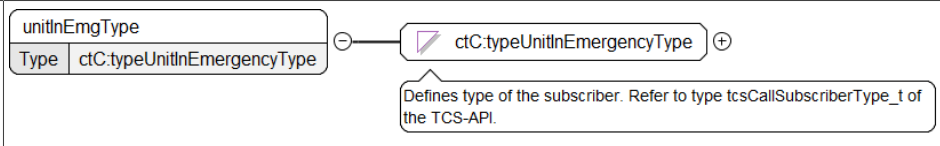
Namespace	DR-GW-Interface/DR-GW-Call.Events
Diagram	
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	`<xs:element name="tetraCallId" type="xs:unsignedLong" minOccurs="0"/>`

Element Call_UnitInEmergencyEvent / unitInEmg

Namespace	DR-GW-Interface/DR-GW-Call.Events

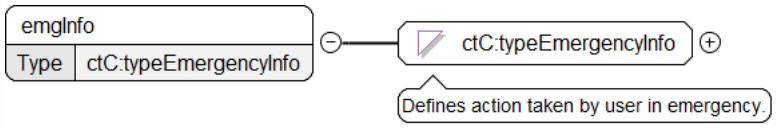
Diagram	
Type	ct:typeAddress
Properties	content: complex
Model	ct:subscriber{0,1} , ct:alias{0,1} , ct:msisdn{0,1} , ct:fssn{0,1} , ct:external{0,1} , ct:opta{0,1} , ct:cell{0,1}
Children	ct:alias, ct:cell, ct:external, ct:fssn, ct:msisdn, ct:opta, ct:subscriber
Instance	<unitInEmg xmlns="DR-GW-Interface/DR-GW-Call.Events" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:subscriber>{0,1}</ct:subscriber> <ct:alias>{0,1}</ct:alias> <ct:msisdn>{0,1}</ct:msisdn> <ct:fssn>{0,1}</ct:fssn> <ct:external>{0,1}</ct:external> <ct:opta>{0,1}</ct:opta> <ct:cell>{0,1}</ct:cell> </unitInEmg>
Source	<xs:element name="unitInEmg" type="ct:typeAddress"/>

Element Call_UnitInEmergencyEvent / unitInEmgType

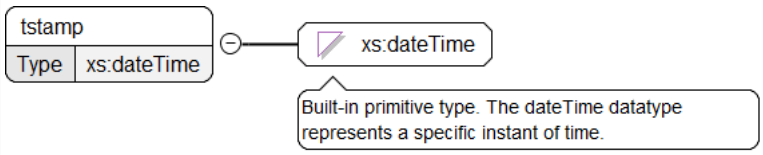
Namespace	DR-GW-Interface/DR-GW-Call.Events
Diagram	
Type	typeUnitInEmergencyType
Properties	content: simple
Facets	enumeration dummy enumeration ms enumeration g4wif enumeration external enumeration ws
Source	<xs:element name="unitInEmgType" type="ctC:typeUnitInEmergencyType"/>

Element Call_UnitInEmergencyEvent / emgInfo

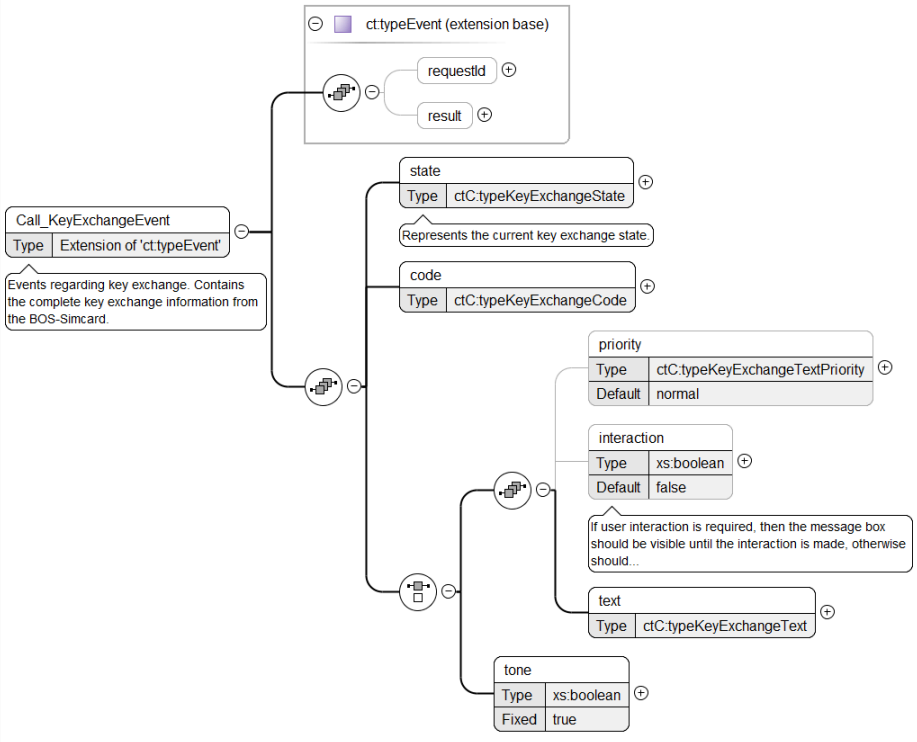
Namespace	DR-GW-Interface/DR-GW-Call.Events

Diagram	
Type	typeEmergencyInfo
Properties	content: simple
Facets	enumeration addTx enumeration add enumeration ceased enumeration demandTx enumeration removed enumeration emergencyCallDisconnected
Source	`<xs:element name="emgInfo" type="ctC:typeEmergencyInfo"/>`

Element Call_UnitInEmergencyEvent / tstamp

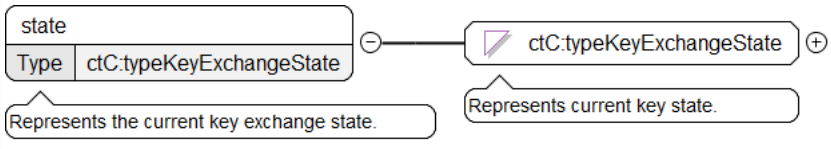
Namespace	DR-GW-Interface/DR-GW-Call.Events
Diagram	
Type	xs:dateTime
Properties	content: simple
Source	`<xs:element name="tstamp" type="xs:dateTime"/>`

Element Call_KeyExchangeEvent

Namespace	DR-GW-Interface/DR-GW-Call.Events
Annotations	Events regarding key exchange. Contains the complete key exchange information from the BOS-Simcard.
Diagram	

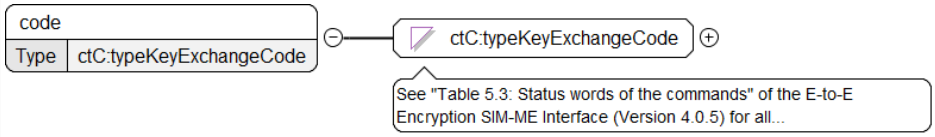
Type	extension of ct:typeEvent
Type hierarchy	• ct:typeEvent
Properties	content: complex
Model	ct:requestId{0,1} , ct:result{0,1} , state , code , ((priority{0,1} , interaction{0,1} , text) | tone)
Children	code, ct:requestId, ct:result, interaction, priority, state, text, tone
Instance	<Call_KeyExchangeEvent xmlns="DR-GW-Interface/DR-GW-Call.Events" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:requestId>{0,1}</ct:requestId> <ct:result>{0,1}</ct:result> <state>{1,1}</state> <code>{1,1}</code> <priority>{0,1}</priority> <interaction>{0,1}</interaction> <text>{1,1}</text> <tone>{1,1}</tone> </Call_KeyExchangeEvent>
Source	<xs:element name="Call_KeyExchangeEvent"> <xs:annotation> <xs:documentation>Events regarding key exchange. Contains the complete key exchange information from the BOS-Simcard.</xs:documentation> </xs:annotation> <xs:complexType> <xs:complexContent> <xs:extension base="ct:typeEvent"> <xs:sequence> <xs:element name="state" type="ctC:typeKeyExchangeState"> <xs:annotation> <xs:documentation>Represents the current key exchange state.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="code" type="ctC:typeKeyExchangeCode"/> <xs:choice> <xs:sequence> <xs:element name="priority" type="ctC:typeKeyExchangeTextPriority" minOccurs="0" default="normal"/> <xs:element name="interaction" type="xs:boolean" minOccurs="0" default="false"> <xs:annotation> <xs:documentation>If user interaction is required, then the message box should be visible until the interaction is made, otherwise should be hidden after delay.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="text" type="ctC:typeKeyExchangeText"/> </xs:sequence> <xs:element name="tone" type="xs:boolean" fixed="true"/> </xs:choice> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </xs:element>

Element Call_KeyExchangeEvent / state

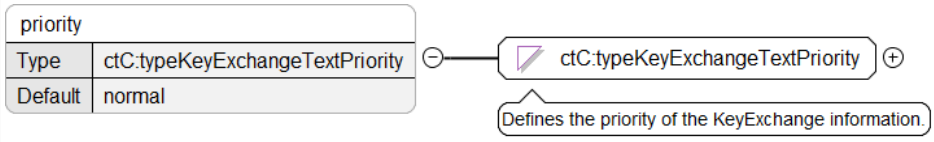
Namespace	DR-GW-Interface/DR-GW-Call.Events		
Annotations	Represents the current key exchange state.		
Diagram			
Type	typeKeyExchangeState		
Properties	content: simple		
Facets	enumeration	keyValid	current key is valid, no user action required.
	enumeration	keyInvalid	Key invalid, user must request key exchange.
	enumeration	keyExchangeInProgress	Key exchange in progress, user may abort exchange or wait until it gets finished.
Source	<xs:element name="state" type="ctC:typeKeyExchangeState">		

```
  <xs:annotation>
    <xs:documentation>Represents the current key exchange state.</xs:documentation>
  </xs:annotation>
</xs:element>
```

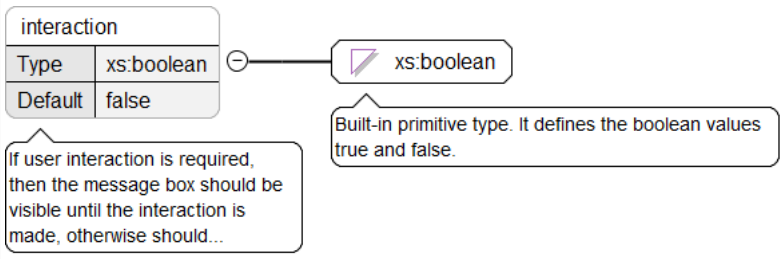
Element Call_KeyExchangeEvent / code

Namespace	DR-GW-Interface/DR-GW-Call.Events
Diagram	
Type	typeKeyExchangeCode
Properties	content: simple
Facets	length 2
Source	`<xs:element name="code" type="ctC:typeKeyExchangeCode"/>`

Element Call_KeyExchangeEvent / priority

Namespace	DR-GW-Interface/DR-GW-Call.Events
Diagram	
Type	typeKeyExchangeTextPriority
Properties	content: simple minOccurs: 0 default: normal
Facets	enumeration normal enumeration high
Source	`<xs:element name="priority" type="ctC:typeKeyExchangeTextPriority" minOccurs="0" default="normal"/>`

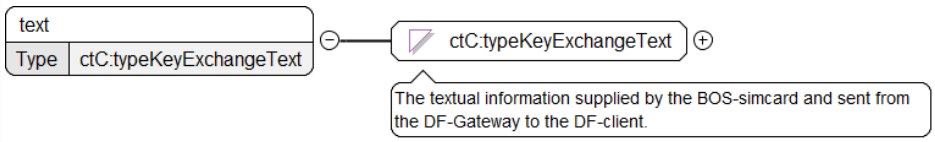
Element Call_KeyExchangeEvent / interaction

Namespace	DR-GW-Interface/DR-GW-Call.Events
Annotations	If user interaction is required, then the message box should be visible until the interaction is made, otherwise should be hidden after delay.
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: false

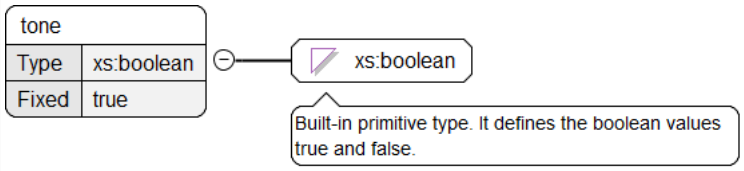
Source	`<xs:element name="interaction" type="xs:boolean" minOccurs="0" default="false">`
	`  <xs:annotation>`
	`  <xs:documentation>If user interaction is required, then the message box should be visible until the interaction is made, otherwise should be hidden after delay.</xs:documentation>`

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

Element `Call_KeyExchangeEvent / text`

Namespace	DR-GW-Interface/DR-GW-Call.Events
Diagram	
Type	typeKeyExchangeText
Properties	content: `simple`
Facets	maxLength `100`
Source	`<xs:element name="text" type="ctC:typeKeyExchangeText"/>`

Element `Call_KeyExchangeEvent / tone`

Namespace	DR-GW-Interface/DR-GW-Call.Events
Diagram	
Type	xs:boolean
Properties	content: `simple` fixed: `true`
Source	`<xs:element name="tone" type="xs:boolean" fixed="true"/>`

Namespace: "DR-GW-Interface/CommonTypes"

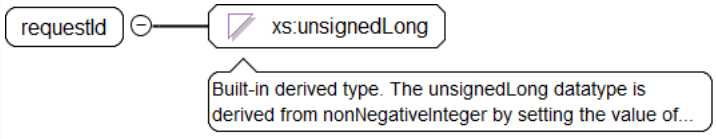
Schema(s)

Imported schema `CommonTypes.xsd`

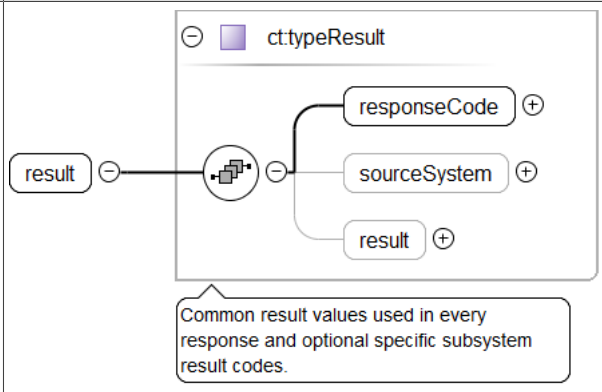
Namespace	DR-GW-Interface/CommonTypes
Annotations	`Version 1.1.1`
Properties	attribute form default: `unqualified` element form default: `qualified`

Element(s)

Element `ct:typeResponse / ct:requestId`

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedLong
Properties	content: `simple`
Source	`<xs:element name="requestId" type="xs:unsignedLong"/>`

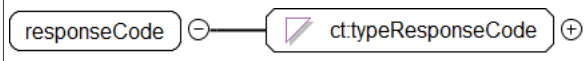
Element `ct:typeResponse / ct:result`

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeResult
Properties	content: complex
Model	ct:responseCode , ct:sourceSystem{0,1} , ct:result{0,1}
Children	ct:responseCode, ct:result, ct:sourceSystem

Instance	`<ct:result xmlns:ct="DR-GW-Interface/CommonTypes">`
	`  <ct:responseCode>{1,1}</ct:responseCode>`
	`  <ct:sourceSystem>{0,1}</ct:sourceSystem>`
	`  <ct:result>{0,1}</ct:result>`
	`</ct:result>`

Source	`<xs:element name="result" type="ct:typeResult"/>`

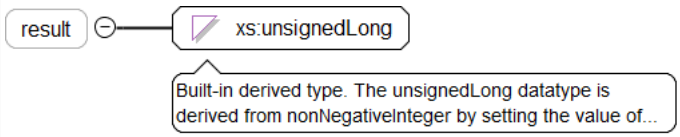
Element `ct:typeResult / ct:responseCode`

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeResponseCode
Properties	content: simple
Facets	enumeration: success enumeration: final_response_pending enumeration: error enumeration: not_authorized_error enumeration: temporary_failure enumeration: subscription_failed
Source	`<xs:element name="responseCode" type="ct:typeResponseCode"/>`

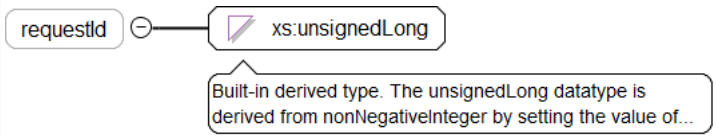
Element `ct:typeResult / ct:sourceSystem`

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeSourceSystem
Properties	content: simple minOccurs: 0
Facets	enumeration: DR-GW enumeration: TCS-API enumeration: TETRA
Source	`<xs:element name="sourceSystem" type="ct:typeSourceSystem" minOccurs="0"/>`

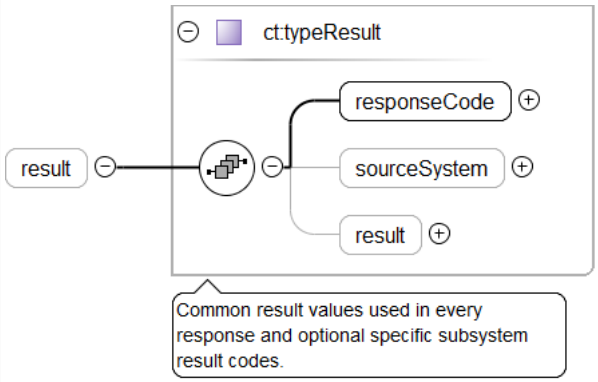
Element ct:typeResult / ct:result

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	`<xs:element name="result" type="xs:unsignedLong" minOccurs="0"/>`

Element ct:typeEvent / ct:requestId

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	`<xs:element name="requestId" type="xs:unsignedLong" minOccurs="0"/>`

Element ct:typeEvent / ct:result

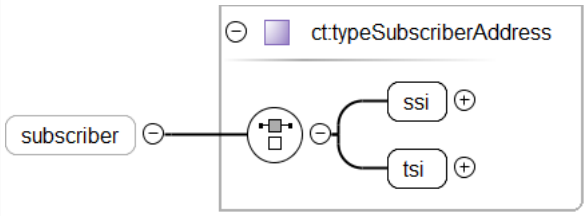
Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeResult
Properties	content: complex minOccurs: 0
Model	ct:responseCode , ct:sourceSystem{0,1} , ct:result{0,1}
Children	ct:responseCode, ct:result, ct:sourceSystem

Instance	`<ct:result xmlns:ct="DR-GW-Interface/CommonTypes">`
	`  <ct:responseCode>{1,1}</ct:responseCode>`
	`  <ct:sourceSystem>{0,1}</ct:sourceSystem>`
	`  <ct:result>{0,1}</ct:result>`
	`</ct:result>`

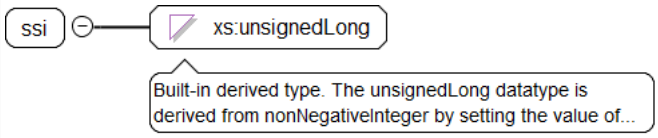
Source	`<xs:element name="result" type="ct:typeResult" minOccurs="0"/>`

Element ct:typeAddress / ct:subscriber

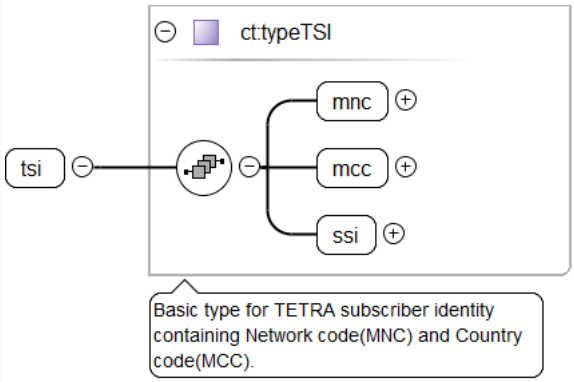
Namespace	DR-GW-Interface/CommonTypes

Diagram	
Type	ct:typeSubscriberAddress
Properties	content: complex minOccurs: 0
Model	ct:ssi | ct:tsi
Children	ct:ssi, ct:tsi
Instance	<ct:subscriber xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:ssi>{1,1}</ct:ssi> <ct:tsi>{1,1}</ct:tsi> </ct:subscriber>
Source	<xs:element name="subscriber" type="ct:typeSubscriberAddress" minOccurs="0"/>

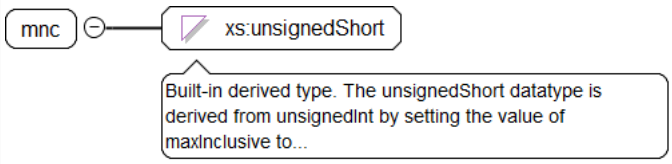
Element ct:typeSubscriberAddress / ct:ssi

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedLong
Properties	content: simple
Source	<xs:element name="ssi" type="xs:unsignedLong"/>

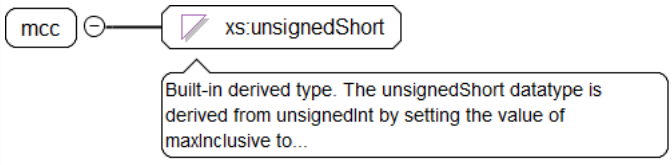
Element ct:typeSubscriberAddress / ct:tsi

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeTSI
Properties	content: complex
Model	ct:mnc , ct:mcc , ct:ssi
Children	ct:mcc, ct:mnc, ct:ssi
Instance	<ct:tsi xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:mnc>{1,1}</ct:mnc> <ct:mcc>{1,1}</ct:mcc> <ct:ssi>{1,1}</ct:ssi> </ct:tsi>
Source	<xs:element name="tsi" type="ct:typeTSI"/>

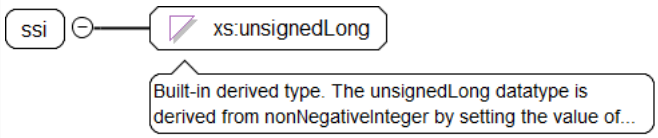
Element ct:typeTSI / ct:mnc

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedShort
Properties	content: simple
Source	`<xs:element name="mnc" type="xs:unsignedShort"/>`

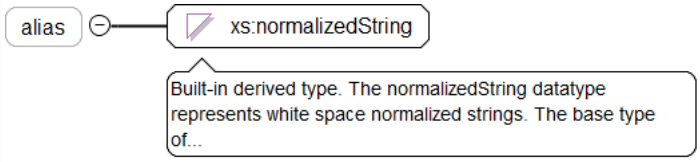
Element ct:typeTSI / ct:mcc

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedShort
Properties	content: simple
Source	`<xs:element name="mcc" type="xs:unsignedShort"/>`

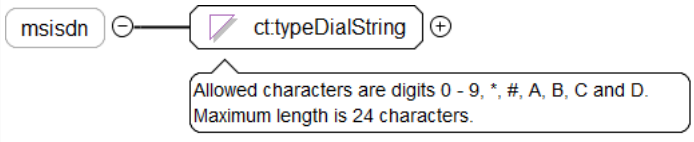
Element ct:typeTSI / ct:ssi

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedLong
Properties	content: simple
Source	`<xs:element name="ssi" type="xs:unsignedLong"/>`

Element ct:typeAddress / ct:alias

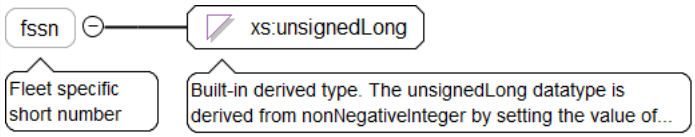
Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:normalizedString
Properties	content: simple minOccurs: 0
Source	`<xs:element name="alias" type="xs:normalizedString" minOccurs="0"/>`

Element ct:typeAddress / ct:msisdn

Namespace	DR-GW-Interface/CommonTypes
Diagram	

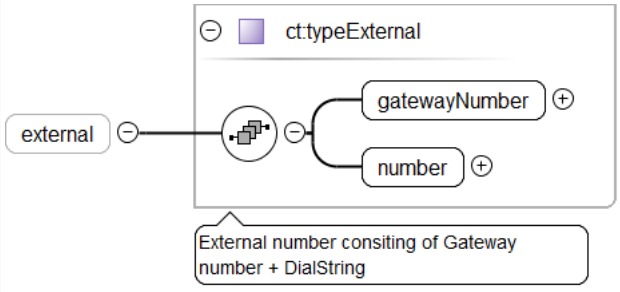
Type	ct:typeDialString
Properties	content: simple minOccurs: 0
Facets	maxLength 24
Source	`<xs:element name="msisdn" type="ct:typeDialString" minOccurs="0"/>`

Element ct:typeAddress / ct:fssn

Namespace	DR-GW-Interface/CommonTypes
Annotations	Fleet specific short number
Diagram	
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0

Source	`<xs:element name="fssn" type="xs:unsignedLong" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation>Fleet specific short number</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

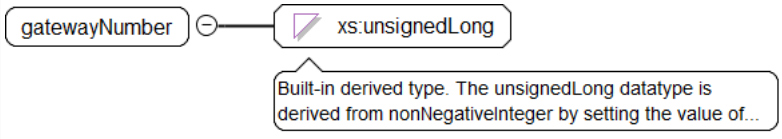
Element ct:typeAddress / ct:external

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeExternal
Properties	content: complex minOccurs: 0
Model	ct:gatewayNumber , ct:number
Children	ct:gatewayNumber, ct:number

Instance	`<ct:external xmlns:ct="DR-GW-Interface/CommonTypes">`
	`  <ct:gatewayNumber>{1,1}</ct:gatewayNumber>`
	`  <ct:number>{1,1}</ct:number>`
	`</ct:external>`

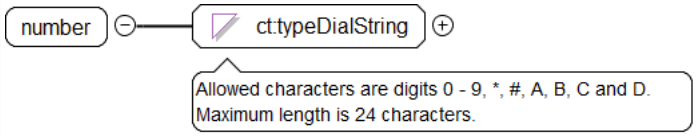
Source	`<xs:element name="external" type="ct:typeExternal" minOccurs="0"/>`

Element ct:typeExternal / ct:gatewayNumber

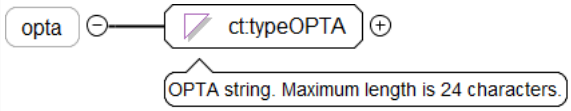
Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedLong

Properties	content:	simple
Source	<xs:element name="gatewayNumber" type="xs:unsignedLong"/>	

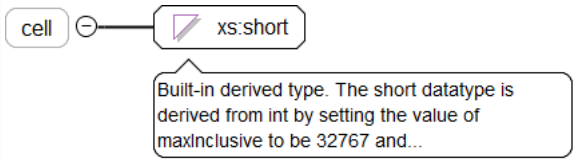
Element ct:typeExternal / ct:number

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	ct:typeDialString
Properties	content: simple
Facets	maxLength 24
Source	<xs:element name="number" type="ct:typeDialString"/>

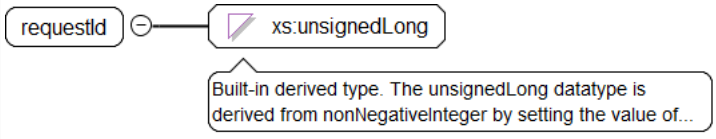
Element ct:typeAddress / ct:opta

Namespace	DR-GW-Interface/CommonTypes	
Diagram		
Type	ct:typeOPTA	
Properties	content:	simple
	minOccurs:	0
Facets	maxLength	24
Source	<xs:element name="opta" type="ct:typeOPTA" minOccurs="0"/>	

Element ct:typeAddress / ct:cell

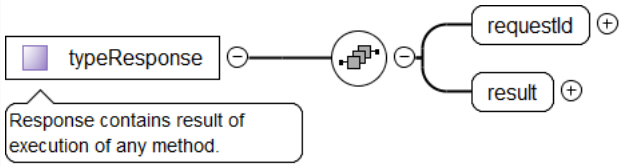
Namespace	DR-GW-Interface/CommonTypes	
Diagram		
Type	xs:short	
Properties	content:	simple
	minOccurs:	0
Source	<xs:element name="cell" type="xs:short" minOccurs="0"/>	

Element ct:typeRequest / ct:requestId

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	xs:unsignedLong
Properties	content: simple
Source	<xs:element name="requestId" type="xs:unsignedLong"/>

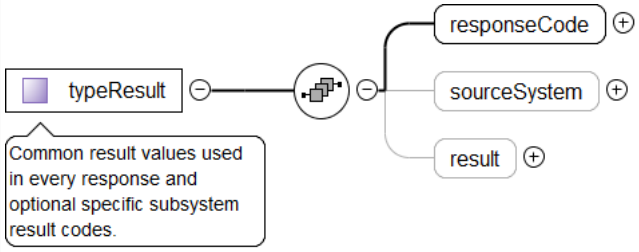
Complex Type(s)

Complex Type `ct:typeResponse`

Namespace	DR-GW-Interface/CommonTypes
Annotations	`Response contains result of execution of any method.`
Diagram	
Used by	Element Call_Response
Model	ct:requestId , ct:result
Children	ct:requestId, ct:result

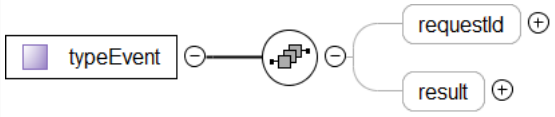
Source	`<xs:complexType name="typeResponse">`
	`  <xs:annotation>`
	`  <xs:documentation>Response contains result of execution of any method.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`  <xs:element name="requestId" type="xs:unsignedLong"/>`
	`  <xs:element name="result" type="ct:typeResult"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `ct:typeResult`

Namespace	DR-GW-Interface/CommonTypes
Annotations	`Common result values used in every response and optional specific subsystem result codes.`
Diagram	
Used by	Elements ct:typeEvent/ct:result, ct:typeResponse/ct:result
Model	ct:responseCode , ct:sourceSystem{0,1} , ct:result{0,1}
Children	ct:responseCode, ct:result, ct:sourceSystem

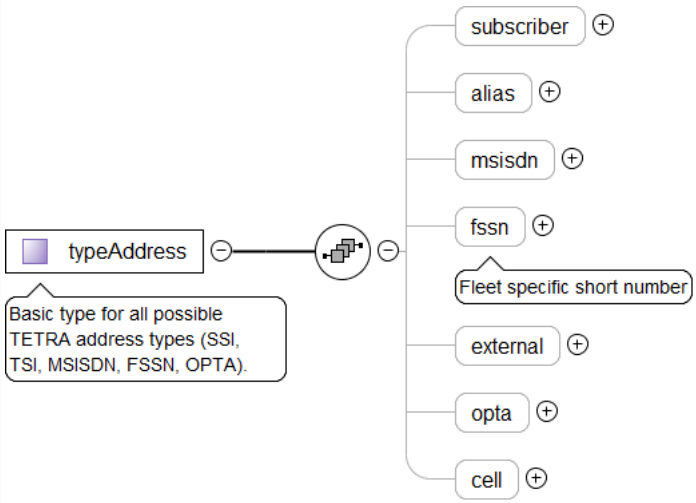
Source	`<xs:complexType name="typeResult">`
	`  <xs:annotation>`
	`  <xs:documentation>Common result values used in every response and optional specific subsystem result codes.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`  <xs:element name="responseCode" type="ct:typeResponseCode"/>`
	`  <xs:element name="sourceSystem" type="ct:typeSourceSystem" minOccurs="0"/>`
	`  <xs:element name="result" type="xs:unsignedLong" minOccurs="0"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `ct:typeEvent`

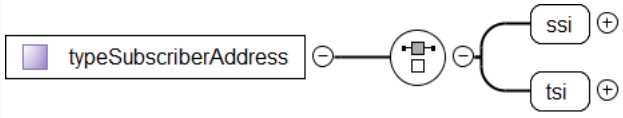
Namespace	DR-GW-Interface/CommonTypes
Diagram	

Used by	Elements Call_Event, Call_KeyExchangeEvent, Call_PTTEvent, Call_SelectEvent, Call_UnitInEmergencyEvent
Model	ct:requestId{0,1} , ct:result{0,1}
Children	ct:requestId, ct:result
Source	<xs:complexType name="typeEvent"> <xs:sequence> <xs:element name="requestId" type="xs:unsignedLong" minOccurs="0"/> <xs:element name="result" type="ct:typeResult" minOccurs="0"/> </xs:sequence> </xs:complexType>

Complex Type ct:typeAddress

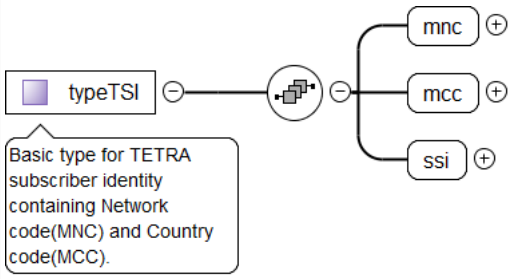
Namespace	DR-GW-Interface/CommonTypes
Annotations	Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).
Diagram	
Used by	Elements Call_Event/calledParty, Call_Event/callingParty, Call_UnitInEmergencyEvent/unitInEmg, typeSelection/target, typeTxGranted/talkingParty
Model	ct:subscriber{0,1} , ct:alias{0,1} , ct:msisdn{0,1} , ct:fssn{0,1} , ct:external{0,1} , ct:opta{0,1} , ct:cell{0,1}
Children	ct:alias, ct:cell, ct:external, ct:fssn, ct:msisdn, ct:opta, ct:subscriber
Source	<xs:complexType name="typeAddress"> <xs:annotation> <xs:documentation>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="subscriber" type="ct:typeSubscriberAddress" minOccurs="0"/> <xs:element name="alias" type="xs:normalizedString" minOccurs="0"/> <xs:element name="msisdn" type="ct:typeDialString" minOccurs="0"/> <xs:element name="fssn" type="xs:unsignedLong" minOccurs="0"> <xs:annotation> <xs:documentation>Fleet specific short number</xs:documentation> </xs:annotation> </xs:element> <xs:element name="external" type="ct:typeExternal" minOccurs="0"/> <xs:element name="opta" type="ct:typeOPTA" minOccurs="0"/> <xs:element name="cell" type="xs:short" minOccurs="0"/> </xs:sequence> </xs:complexType>

Complex Type ct:typeSubscriberAddress

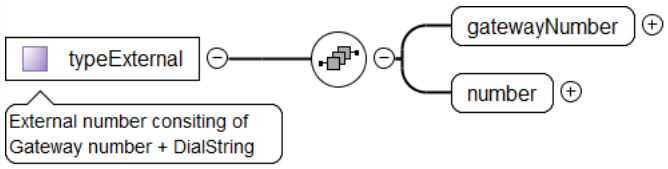
Namespace	DR-GW-Interface/CommonTypes
Annotations	
Diagram	

Used by	Elements Call_UnitInEmergencyEvent/group, ct:typeAddress/ct:subscriber
Model	ct:ssi | ct:tsi
Children	ct:ssi, ct:tsi
Source	<xs:complexType name="typeSubscriberAddress"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:choice> <xs:element name="ssi" type="xs:unsignedLong"/> <xs:element name="tsi" type="ct:typeTSI"/> </xs:choice> </xs:complexType>

Complex Type `ct:typeTSI`

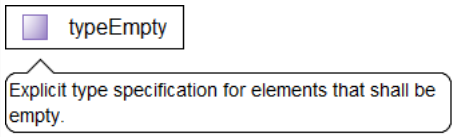
Namespace	DR-GW-Interface/CommonTypes
Annotations	Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC).
Diagram	
Used by	Element ct:typeSubscriberAddress/ct:tsi
Model	ct:mnc , ct:mcc , ct:ssi
Children	ct:mcc, ct:mnc, ct:ssi
Source	<xs:complexType name="typeTSI"> <xs:annotation> <xs:documentation>Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC).</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="mnc" type="xs:unsignedShort"/> <xs:element name="mcc" type="xs:unsignedShort"/> <xs:element name="ssi" type="xs:unsignedLong"/> </xs:sequence> </xs:complexType>

Complex Type `ct:typeExternal`

Namespace	DR-GW-Interface/CommonTypes
Annotations	External number consiting of Gateway number + DialString
Diagram	
Used by	Element ct:typeAddress/ct:external
Model	ct:gatewayNumber , ct:number
Children	ct:gatewayNumber, ct:number
Source	<xs:complexType name="typeExternal"> <xs:annotation> <xs:documentation>External number consiting of Gateway number + DialString</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="gatewayNumber" type="xs:unsignedLong"/> <xs:element name="number" type="ct:typeDialString"/> </xs:sequence>

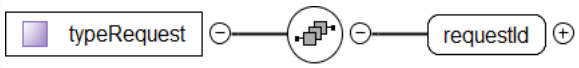
```
</xs:complexType>
```

Complex Type `ct:typeEmpty`

Namespace	DR-GW-Interface/CommonTypes
Annotations	Explicit type specification for elements that shall be empty.
Diagram	
Used by	Elements Call_PTTEvent/ceased, Call_PTTEvent/wait

Source	`<xs:complexType name="typeEmpty">`
	`  <xs:annotation>`
	`  <xs:documentation>Explicit type specification for elements that shall be empty.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:complexType>`

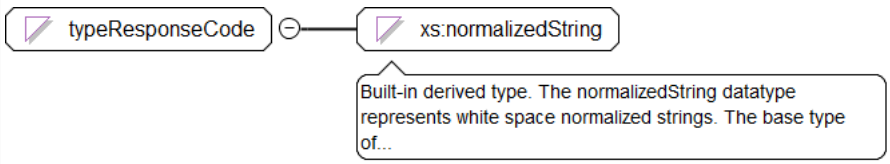
Complex Type `ct:typeRequest`

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Model	ct:requestId
Children	ct:requestId

Source	`<xs:complexType name="typeRequest">`
	`  <xs:sequence>`
	`  <xs:element name="requestId" type="xs:unsignedLong"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

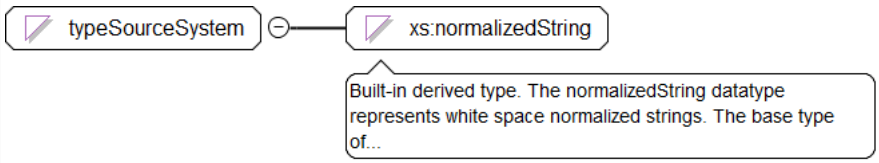
Simple Type(s)

Simple Type `ct:typeResponseCode`

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `success` enumeration `final_response_pending` enumeration `error` enumeration `not_authorized_error` enumeration `temporary_failure` enumeration `subscription_failed`
Used by	Element ct:typeResult/ct:responseCode

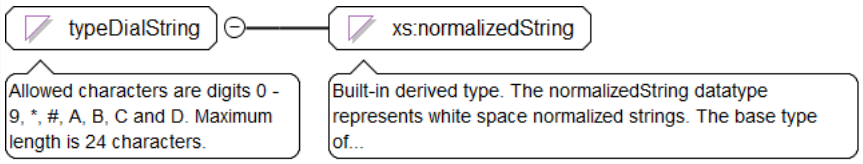
Source	`<xs:simpleType name="typeResponseCode">`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="success"/>`
	`  <xs:enumeration value="final_response_pending"/>`
	`  <xs:enumeration value="error"/>`
	`  <xs:enumeration value="not_authorized_error"/>`
	`  <xs:enumeration value="temporary_failure"/>`
	`  <xs:enumeration value="subscription_failed"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `ct:typeSourceSystem`

Namespace	DR-GW-Interface/CommonTypes
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration DR-GW enumeration TCS-API enumeration TETRA
Used by	Element ct:typeResult/ct:sourceSystem

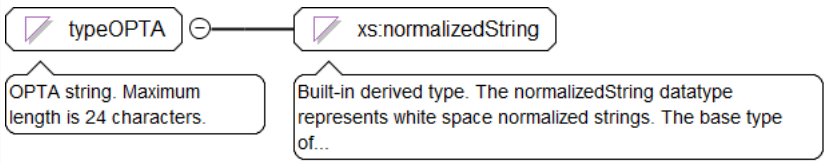
Source	`<xs:simpleType name="typeSourceSystem">`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="DR-GW"/>`
	`  <xs:enumeration value="TCS-API"/>`
	`  <xs:enumeration value="TETRA"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `ct:typeDialString`

Namespace	DR-GW-Interface/CommonTypes
Annotations	Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.
Diagram	
Type	restriction of xs:normalizedString
Facets	maxLength 24
Used by	Elements ct:typeAddress/ct:msisdn, ct:typeExternal/ct:number

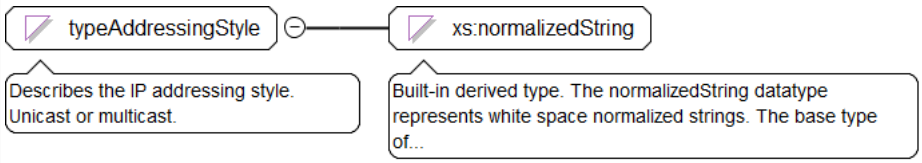
Source	`<xs:simpleType name="typeDialString">`
	`  <xs:annotation>`
	`  <xs:documentation>Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:maxLength value="24"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `ct:typeOPTA`

Namespace	DR-GW-Interface/CommonTypes
Annotations	OPTA string. Maximum length is 24 characters.
Diagram	
Type	restriction of xs:normalizedString
Facets	maxLength 24
Used by	Element ct:typeAddress/ct:opta
Source	`<xs:simpleType name="typeOPTA">`

```
  <xs:annotation>
    <xs:documentation>OPTA string. Maximum length is 24 characters.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:maxLength value="24"/>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `ct:typeAddressingStyle`

Namespace	DR-GW-Interface/CommonTypes
Annotations	Describes the IP addressing style. Unicast or multicast.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration: ucast enumeration: mcast

Source	`<xs:simpleType name="typeAddressingStyle">`
	`  <xs:annotation>`
	`  <xs:documentation>Describes the IP addressing style. Unicast or multicast.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="ucast"/>`
	`  <xs:enumeration value="mcast"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Namespace: "DR-GW-Interface/DR-GW-Call.CommonTypes"

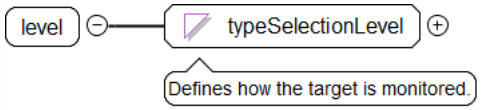
Schema(s)

Imported schema `DR-GW-Call.CommonTypes.xsd`

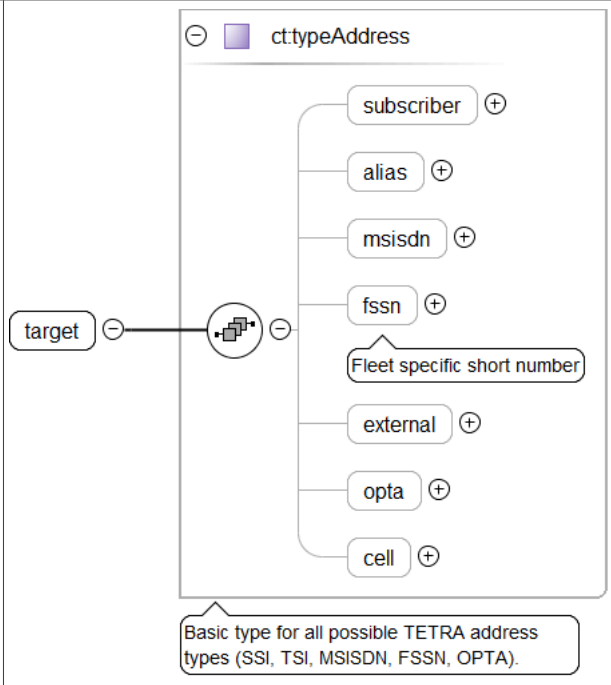
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	Version 1.1.1
Properties	attribute form default: unqualified element form default: qualified

Element(s)

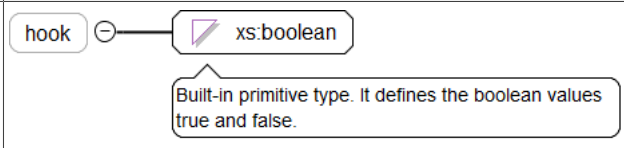
Element `typeSelection / level`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	
Type	typeSelectionLevel
Properties	content: simple
Facets	enumeration: no — No selection. Used to remove selection. enumeration: event — Event monitoring. enumeration: audio — Audio monitoring. enumeration: use — Selection level use. enumeration: a_use — Selection level active use.
Source	`<xs:element name="level" type="typeSelectionLevel"/>`

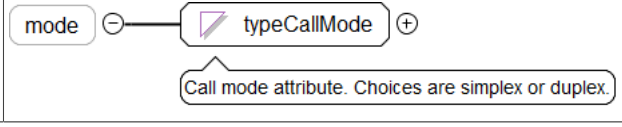
Element typeSelection / target

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	
Type	ct:typeAddress
Properties	content: complex
Model	ct:subscriber{0,1} , ct:alias{0,1} , ct:msisdn{0,1} , ct:fssn{0,1} , ct:external{0,1} , ct:opta{0,1} , ct:cell{0,1}
Children	ct:alias, ct:cell, ct:external, ct:fssn, ct:msisdn, ct:opta, ct:subscriber
Instance	<target xmlns="DR-GW-Interface/DR-GW-Call.CommonTypes" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:subscriber>{0,1}</ct:subscriber> <ct:alias>{0,1}</ct:alias> <ct:msisdn>{0,1}</ct:msisdn> <ct:fssn>{0,1}</ct:fssn> <ct:external>{0,1}</ct:external> <ct:opta>{0,1}</ct:opta> <ct:cell>{0,1}</ct:cell> </target>
Source	<xs:element name="target" type="ct:typeAddress"/>

Element typeCallAttributes / hook

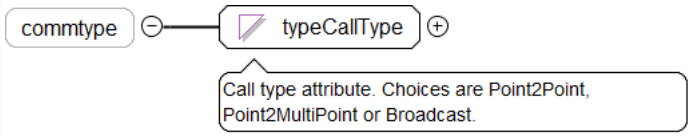
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0
Source	<xs:element name="hook" type="xs:boolean" minOccurs="0"/>

Element typeCallAttributes / mode

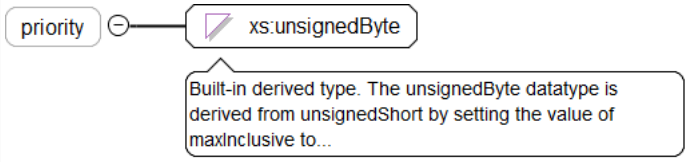
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	

Type	typeCallMode
Properties	content: simple minOccurs: 0
Facets	enumeration simplex enumeration duplex
Source	`<xs:element name="mode" type="typeCallMode" minOccurs="0"/>`

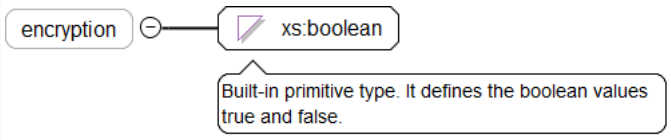
Element typeCallAttributes / commtype

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	
Type	typeCallType
Properties	content: simple minOccurs: 0
Facets	enumeration p2p enumeration p2mp enumeration bcast
Source	`<xs:element name="commtype" type="typeCallType" minOccurs="0"/>`

Element typeCallAttributes / priority

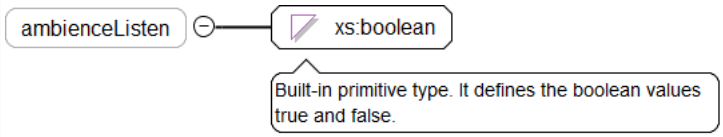
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	
Type	xs:unsignedByte
Properties	content: simple minOccurs: 0 default: 1
Source	`<xs:element name="priority" type="xs:unsignedByte" default="1" minOccurs="0"/>`

Element typeCallAttributes / encryption

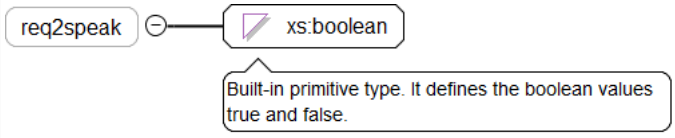
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: true
Source	`<xs:element name="encryption" type="xs:boolean" default="true" minOccurs="0"/>`

Element typeCallAttributes / ambienceListen

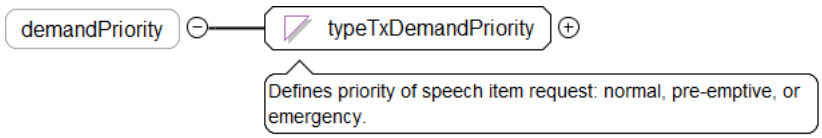
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes

Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: 0
Source	`<xs:element name="ambienceListen" type="xs:boolean" default="0" minOccurs="0"/>`

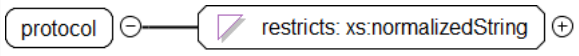
Element typeCallAttributes / req2speak

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: 1
Source	`<xs:element name="req2speak" type="xs:boolean" default="1" minOccurs="0"/>`

Element typeCallAttributes / demandPriority

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	
Type	typeTxDemandPriority
Properties	content: simple minOccurs: 0 default: normal
Facets	enumeration normal enumeration preemptive enumeration emergency
Source	`<xs:element name="demandPriority" type="typeTxDemandPriority" default="normal" minOccurs="0"/>`

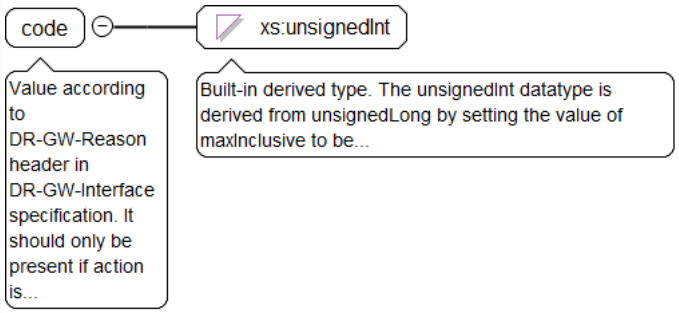
Element typeDisconnectCause / protocol

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	
Type	restriction of xs:normalizedString
Properties	content: simple
Facets	enumeration DR-GW enumeration TCS-API

Source	`<xs:element name="protocol">`
	`  <xs:simpleType>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="DR-GW"/>`
	`  <xs:enumeration value="TCS-API"/>`

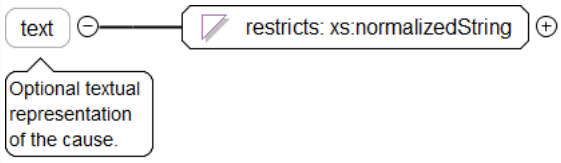
```
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Element typeDisconnectCause / code

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	Value according to DR-GW-Reason header in DR-GW-Interface specification. It should only be present if action is "disconnected" and holds the reason for call disconnection.
Diagram	
Type	xs:unsignedInt
Properties	content: simple
Source	

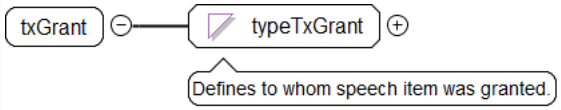
```
<xs:element name="code" type="xs:unsignedInt">
  <xs:annotation>
    <xs:documentation>Value according to DR-GW-Reason header in DR-GW-Interface specification. It
 should only be present if action is "disconnected" and holds the reason for call disconnection.</
xs:documentation>
  </xs:annotation>
</xs:element>
```

Element typeDisconnectCause / text

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	Optional textual representation of the cause.
Diagram	
Type	restriction of xs:normalizedString
Properties	content: simple minOccurs: 0
Facets	maxLength 80
Source	

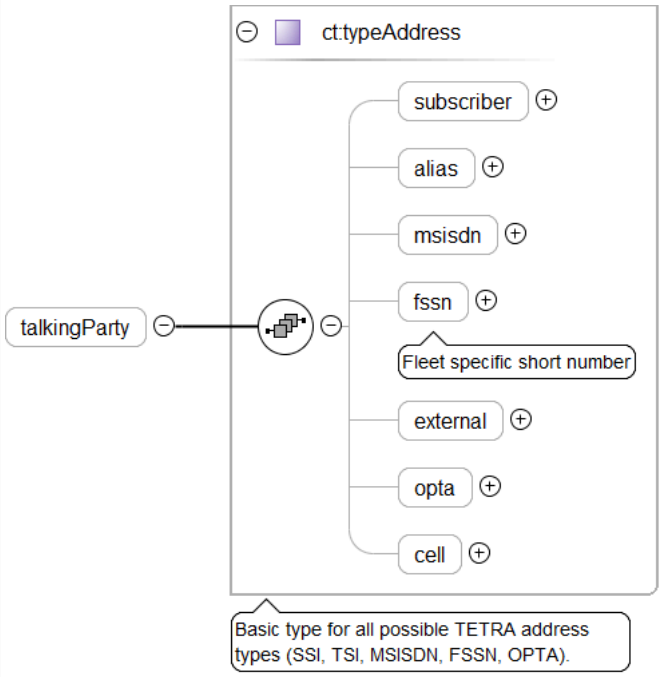
```
<xs:element name="text" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Optional textual representation of the cause.</xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:normalizedString">
      <xs:maxLength value="80"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Element typeTxGranted / txGrant

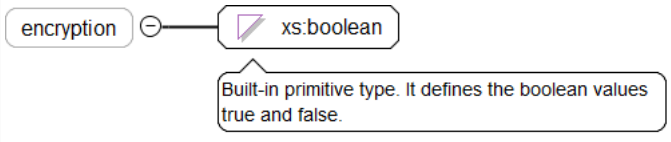
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	
Type	typeTxGrant

Properties	content:	simple
Facets	enumeration	granted
	enumeration	notGranted
	enumeration	queued
	enumeration	granted2another
Source	<xs:element name="txGrant" type="typeTxGrant"/>	

Element typeTxGranted / talkingParty

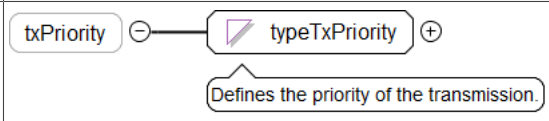
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	
Type	ct:typeAddress
Properties	content: complex minOccurs: 0
Model	ct:subscriber{0,1} , ct:alias{0,1} , ct:msisdn{0,1} , ct:fssn{0,1} , ct:external{0,1} , ct:opta{0,1} , ct:cell{0,1}
Children	ct:alias, ct:cell, ct:external, ct:fssn, ct:msisdn, ct:opta, ct:subscriber
Instance	<talkingParty xmlns="DR-GW-Interface/DR-GW-Call.CommonTypes" xmlns:ct="DR-GW-Interface/CommonTypes"> <ct:subscriber>{0,1}</ct:subscriber> <ct:alias>{0,1}</ct:alias> <ct:msisdn>{0,1}</ct:msisdn> <ct:fssn>{0,1}</ct:fssn> <ct:external>{0,1}</ct:external> <ct:opta>{0,1}</ct:opta> <ct:cell>{0,1}</ct:cell> </talkingParty>
Source	<xs:element name="talkingParty" type="ct:typeAddress" minOccurs="0"/>

Element typeTxGranted / encryption

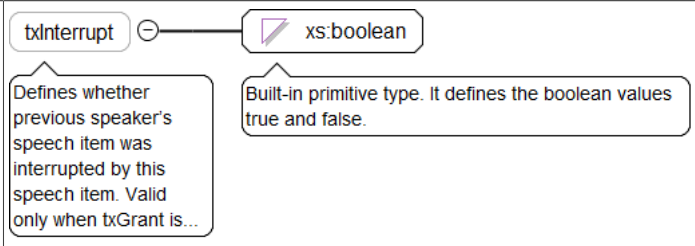
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	
Type	xs:boolean
Properties	content: simple

	minOccurs:	0
	default:	true
Source	<xs:element name="encryption" type="xs:boolean" default="true" minOccurs="0"/>	

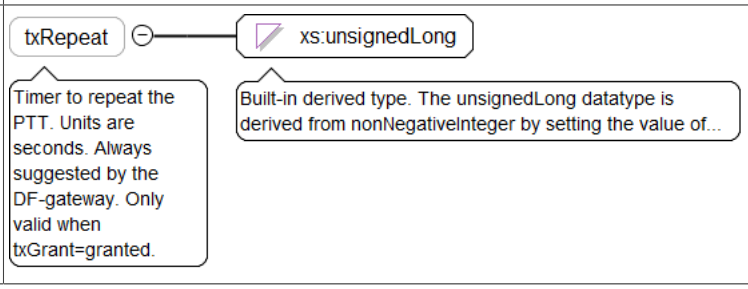
Element typeTxGranted / txPriority

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	
Type	typeTxPriority
Properties	content: simple minOccurs: 0 default: normal
Facets	enumeration normal enumeration emergency
Source	<xs:element name="txPriority" type="typeTxPriority" minOccurs="0" default="normal"/>

Element typeTxGranted / txInterrupt

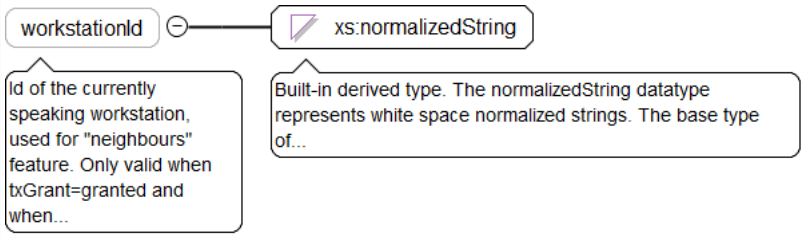
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	Defines whether previous speaker's speech item was interrupted by this speech item. Valid only when txGrant is granted2another.
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: false
Source	<xs:element name="txInterrupt" type="xs:boolean" default="false" minOccurs="0"> <xs:annotation> <xs:documentation>Defines whether previous speaker's speech item was interrupted by this speech item. Valid only when txGrant is granted2another.</xs:documentation> </xs:annotation> </xs:element>

Element typeTxGranted / txRepeat

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	Timer to repeat the PTT. Units are seconds. Always suggested by the DF-gateway. Only valid when txGrant=granted.
Diagram	

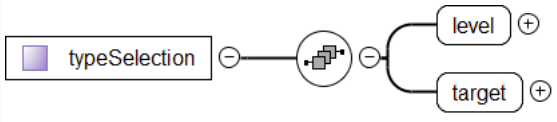
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0 default: 0
Source	<xs:element name="txRepeat" type="xs:unsignedLong" minOccurs="0" default="0"> <xs:annotation> <xs:documentation>Timer to repeat the PTT. Units are seconds. Always suggested by the DF-gateway. Only valid when txGrant=granted.</xs:documentation> </xs:annotation> </xs:element>

Element typeTxGranted / workstationId

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	Id of the currently speaking workstation, used for "neighbours" feature. Only valid when txGrant=granted and when supplied by the DF-client in PTT request.
Diagram	
Type	xs:normalizedString
Properties	content: simple minOccurs: 0
Source	<xs:element name="workstationId" type="xs:normalizedString" minOccurs="0"> <xs:annotation> <xs:documentation>Id of the currently speaking workstation, used for "neighbours" feature. Only valid when txGrant=granted and when supplied by the DF-client in PTT request.</xs:documentation> </xs:annotation> </xs:element>

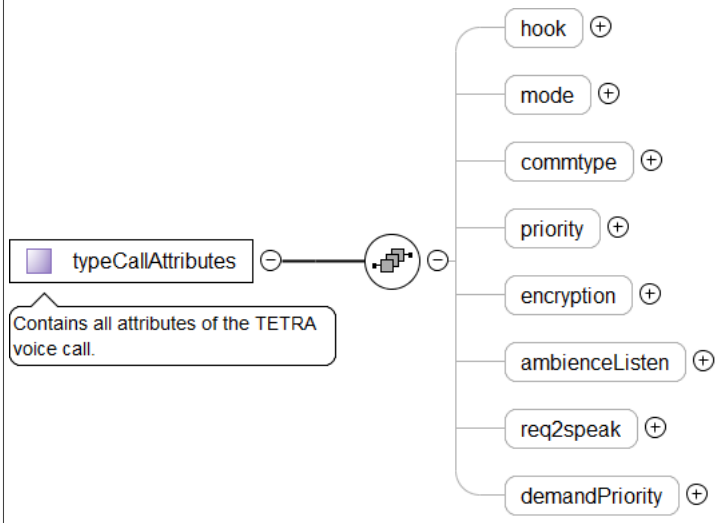
Complex Type(s)

Complex Type typeSelection

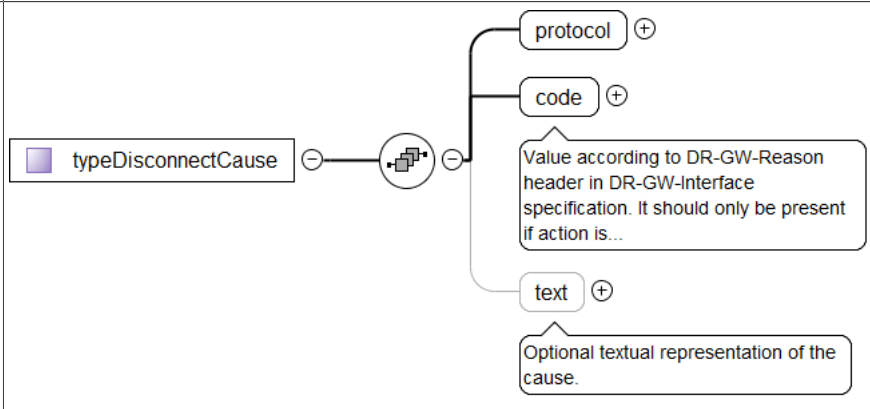
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	
Diagram	
Used by	Element Call_SelectEvent/sel
Model	level , target
Children	level, target
Source	<xs:complexType name="typeSelection"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="level" type="typeSelectionLevel"/> <xs:element name="target" type="ct:typeAddress"/> </xs:sequence> </xs:complexType>

Complex Type typeCallAttributes

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes

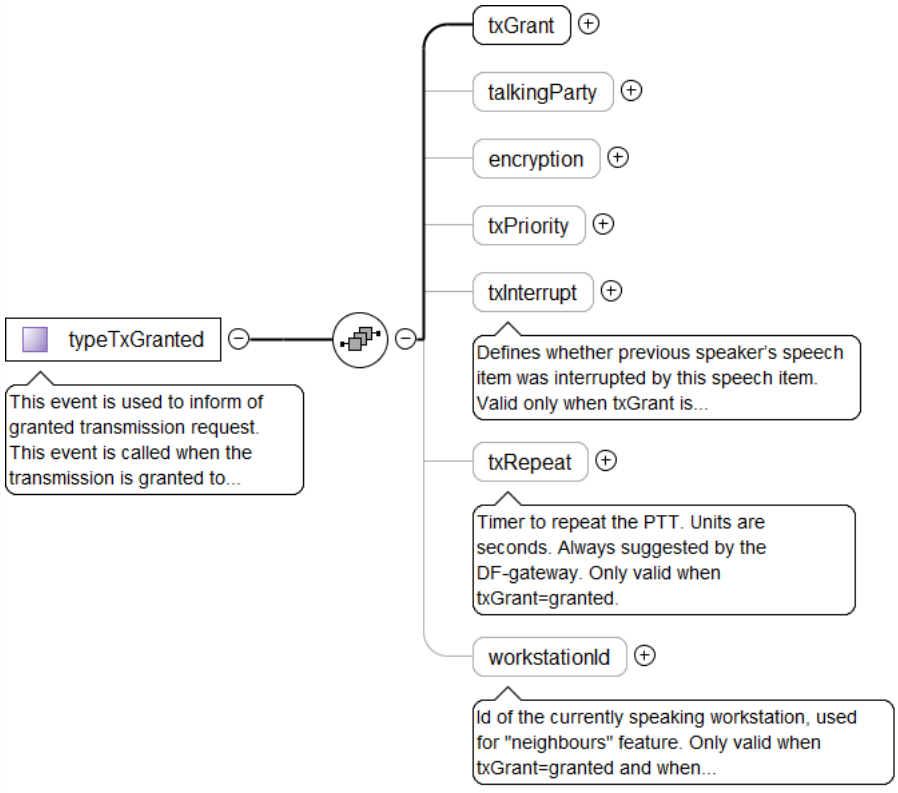
Annotations	Contains all attributes of the TETRA voice call.
Diagram	
Used by	Element Call_Event/attributes
Model	hook{0,1} , mode{0,1} , commtype{0,1} , priority{0,1} , encryption{0,1} , ambienceListen{0,1} , req2speak{0,1} , demandPriority{0,1}
Children	ambienceListen, commtype, demandPriority, encryption, hook, mode, priority, req2speak
Source	<xs:complexType name="typeCallAttributes"> <xs:annotation> <xs:documentation>Contains all attributes of the TETRA voice call.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="hook" type="xs:boolean" minOccurs="0"/> <xs:element name="mode" type="typeCallMode" minOccurs="0"/> <xs:element name="commtype" type="typeCallType" minOccurs="0"/> <xs:element name="priority" type="xs:unsignedByte" default="1" minOccurs="0"/> <xs:element name="encryption" type="xs:boolean" default="true" minOccurs="0"/> <xs:element name="ambienceListen" type="xs:boolean" default="0" minOccurs="0"/> <xs:element name="req2speak" type="xs:boolean" default="1" minOccurs="0"/> <xs:element name="demandPriority" type="typeTxDemandPriority" default="normal" minOccurs="0"/> </xs:sequence> </xs:complexType>

Complex Type typeDisconnectCause

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Diagram	
Used by	Element Call_Event/disconnectCause
Model	protocol , code , text{0,1}
Children	code, protocol, text
Source	<xs:complexType name="typeDisconnectCause"> <xs:sequence> <xs:element name="protocol"> <xs:simpleType> <xs:restriction base="xs:normalizedString">

```
          <xs:enumeration value="DR-GW"/>
          <xs:enumeration value="TCS-API"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="code" type="xs:unsignedInt">
      <xs:annotation>
        <xs:documentation>Value according to DR-GW-Reason header in DR-GW-Interface specification.
 It should only be present if action is "disconnected" and holds the reason for call
 disconnection.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="text" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Optional textual representation of the cause.</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
        <xs:restriction base="xs:normalizedString">
          <xs:maxLength value="80"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

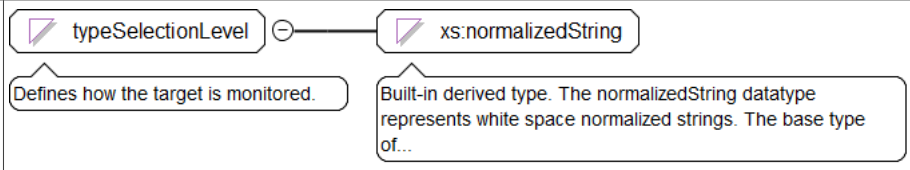
Complex Type `typeTxGranted`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	This event is used to inform of granted transmission request. This event is called when the transmission is granted to client, to another user or the request has been queued.
Diagram	
Used by	Element Call_PTTEvent/granted
Model	txGrant , talkingParty{0,1} , encryption{0,1} , txPriority{0,1} , txInterrupt{0,1} , txRepeat{0,1} , workstationId{0,1}
Children	encryption, talkingParty, txGrant, txInterrupt, txPriority, txRepeat, workstationId
Source	<xs:complexType name="typeTxGranted"> <xs:annotation> <xs:documentation>This event is used to inform of granted transmission request. This event is called when the transmission is granted to client, to another user or the request has been queued.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="txGrant" type="typeTxGrant"/>

```
    <xs:element name="talkingParty" type="ct:typeAddress" minOccurs="0"/>
    <xs:element name="encryption" type="xs:boolean" default="true" minOccurs="0"/>
    <xs:element name="txPriority" type="typeTxPriority" minOccurs="0" default="normal"/>
    <xs:element name="txInterrupt" type="xs:boolean" default="false" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Defines whether previous speaker's speech item was interrupted by this
 speech item. Valid only when txGrant is granted2another.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="txRepeat" type="xs:unsignedLong" minOccurs="0" default="0">
      <xs:annotation>
        <xs:documentation>Timer to repeat the PTT. Units are seconds. Always suggested by the DF-
gateway. Only valid when txGrant=granted.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="workstationId" type="xs:normalizedString" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Id of the currently speaking workstation, used for "neighbours"
 feature. Only valid when txGrant=granted and when supplied by the DF-client in PTT request.</
xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Simple Type(s)

Simple Type `typeSelectionLevel`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	`Defines how the target is monitored.`
Diagram	
Type	restriction of xs:normalizedString

Facets

enumeration	`no`	`No selection. Used to remove selection.`
enumeration	`event`	`Event monitoring.`
enumeration	`audio`	`Audio monitoring.`
enumeration	`use`	`Selection level use.`
enumeration	`a_use`	`Selection level active use.`

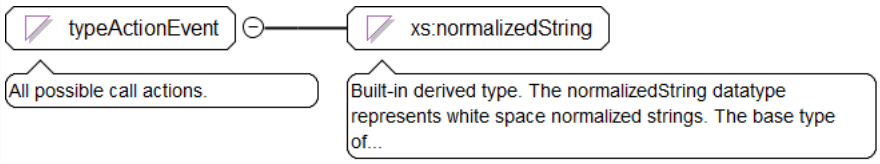
Used by	Element	typeSelection/level

Source

```
<xs:simpleType name="typeSelectionLevel">
  <xs:annotation>
    <xs:documentation>Defines how the target is monitored.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="no">
      <xs:annotation>
        <xs:documentation>No selection. Used to remove selection.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="event">
      <xs:annotation>
        <xs:documentation>Event monitoring.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="audio">
      <xs:annotation>
        <xs:documentation>Audio monitoring.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="use">
      <xs:annotation>
        <xs:documentation>Selection level use.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="a_use">
      <xs:annotation>
        <xs:documentation>Selection level active use.</xs:documentation>
```

```
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `typeActionEvent`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	`All possible call actions.`
Diagram	
Type	restriction of xs:normalizedString

Facets

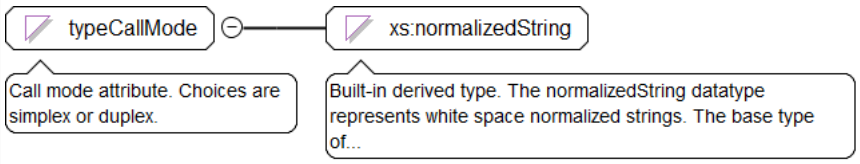
enumeration	`incoming`	`This event fired when there is an incoming call. This is the first indication of a new incoming call.`
enumeration	`connected`	`This event is used to inform that call has been connected and call setup is finished.`
enumeration	`held`	`This event is used to inform TCS Client that individual call was put to hold.`
enumeration	`resumed`	`This event is used to inform that individual call has been taken from hold.`
enumeration	`disconnected`	`This event is used to inform that the call was disconnected.`
enumeration	`transferred`	`This event is a response to transfer method call and indicates the result of the request.`

Used by	Element	Call_Event/action

Source

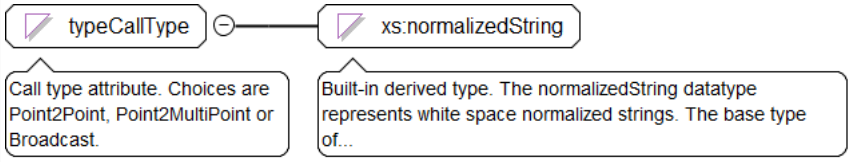
```
<xs:simpleType name="typeActionEvent">
  <xs:annotation>
    <xs:documentation>All possible call actions.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="incoming">
      <xs:annotation>
        <xs:documentation>This event fired when there is an incoming call. This is the first
 indication of a new incoming call.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="connected">
      <xs:annotation>
        <xs:documentation>This event is used to inform that call has been connected and call setup
 is finished.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="held">
      <xs:annotation>
        <xs:documentation>This event is used to inform TCS Client that individual call was put to
 hold.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="resumed">
      <xs:annotation>
        <xs:documentation>This event is used to inform that individual call has been taken from
 hold.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="disconnected">
      <xs:annotation>
        <xs:documentation>This event is used to inform that the call was disconnected.</
xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="transferred">
      <xs:annotation>
        <xs:documentation>This event is a response to transfer method call and indicates the result
 of the request.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `typeCallMode`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	`Call mode attribute. Choices are simplex or duplex.`
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `simplex` enumeration `duplex`
Used by	Element typeCallAttributes/mode

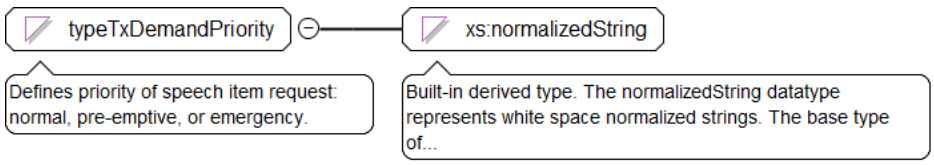
Source	`<xs:simpleType name="typeCallMode">`
	`  <xs:annotation>`
	`  <xs:documentation>Call mode attribute. Choices are simplex or duplex.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="simplex"/>`
	`  <xs:enumeration value="duplex"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `typeCallType`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	`Call type attribute. Choices are Point2Point, Point2MultiPoint or Broadcast.`
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `p2p` enumeration `p2mp` enumeration `bcast`
Used by	Element typeCallAttributes/commtype

Source	`<xs:simpleType name="typeCallType">`
	`  <xs:annotation>`
	`  <xs:documentation>Call type attribute. Choices are Point2Point, Point2MultiPoint or Broadcast.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="p2p"/>`
	`  <xs:enumeration value="p2mp"/>`
	`  <xs:enumeration value="bcast"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

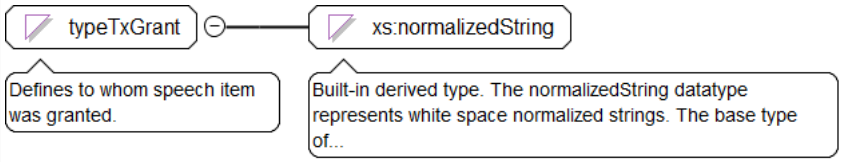
Simple Type `typeTxDemandPriority`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	`Defines priority of speech item request: normal, pre-emptive, or emergency.`
Diagram	

Type	restriction of xs:normalizedString
Facets	enumeration `normal` enumeration `preemptive` enumeration `emergency`
Used by	Element typeCallAttributes/demandPriority

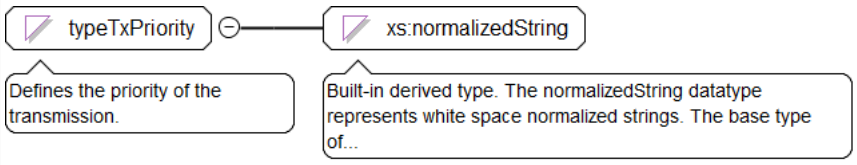
Source	`<xs:simpleType name="typeTxDemandPriority">`
	`  <xs:annotation>`
	`  <xs:documentation>Defines priority of speech item request: normal, pre-emptive, or emergency.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="normal"/>`
	`  <xs:enumeration value="preemptive"/>`
	`  <xs:enumeration value="emergency"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `typeTxGrant`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	`Defines to whom speech item was granted.`
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `granted` enumeration `notGranted` enumeration `queued` enumeration `granted2another`
Used by	Element typeTxGranted/txGrant

Source	`<xs:simpleType name="typeTxGrant">`
	`  <xs:annotation>`
	`  <xs:documentation>Defines to whom speech item was granted.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="granted"/>`
	`  <xs:enumeration value="notGranted"/>`
	`  <xs:enumeration value="queued"/>`
	`  <xs:enumeration value="granted2another"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

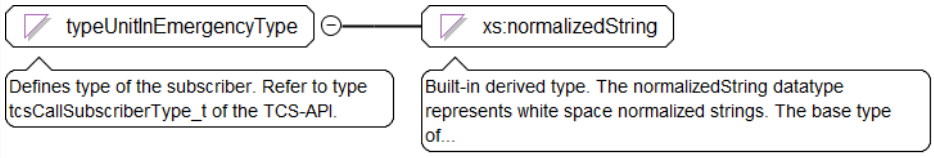
Simple Type `typeTxPriority`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	`Defines the priority of the transmission.`
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `normal` enumeration `emergency`
Used by	Element typeTxGranted/txPriority

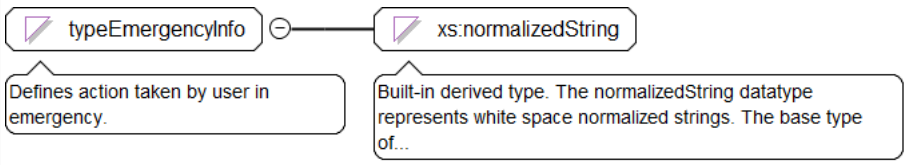
Source	`<xs:simpleType name="typeTxPriority">`
	`  <xs:annotation>`
	`  <xs:documentation>Defines the priority of the transmission.</xs:documentation>`

```
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="normal"/>
    <xs:enumeration value="emergency"/>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `typeUnitInEmergencyType`

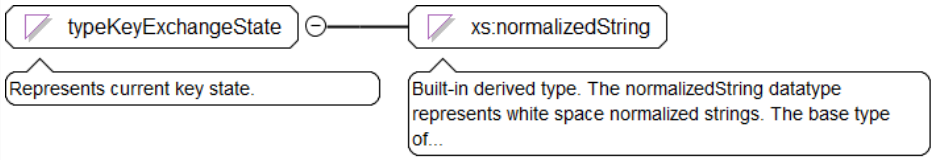
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	Defines type of the subscriber. Refer to type tcsCallSubscriberType_t of the TCS-API.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration: dummy enumeration: ms enumeration: g4wif enumeration: external enumeration: ws
Used by	Element: Call_UnitInEmergencyEvent/unitInEmgType
Source	<xs:simpleType name="typeUnitInEmergencyType"> <xs:annotation> <xs:documentation>Defines type of the subscriber. Refer to type tcsCallSubscriberType_t of the TCS-API.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="dummy"/> <xs:enumeration value="ms"/> <xs:enumeration value="g4wif"/> <xs:enumeration value="external"/> <xs:enumeration value="ws"/> </xs:restriction> </xs:simpleType>

Simple Type `typeEmergencyInfo`

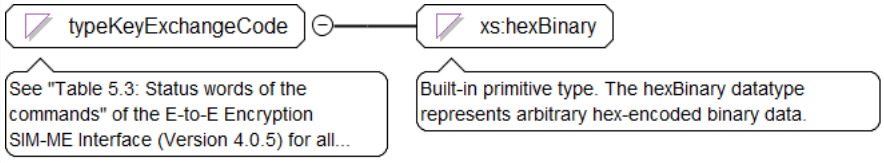
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	Defines action taken by user in emergency.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration: addTx enumeration: add enumeration: ceased enumeration: demandTx enumeration: removed enumeration: emergencyCallDisconnected
Used by	Element: Call_UnitInEmergencyEvent/emgInfo
Source	<xs:simpleType name="typeEmergencyInfo"> <xs:annotation> <xs:documentation>Defines action taken by user in emergency.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="addTx"/>

```
    <xs:enumeration value="add"/>
    <xs:enumeration value="ceased"/>
    <xs:enumeration value="demandTx"/>
    <xs:enumeration value="removed"/>
    <xs:enumeration value="emergencyCallDisconnected"/>
  </xs:restriction>
</xs:simpleType>
```

Simple Type typeKeyExchangeState

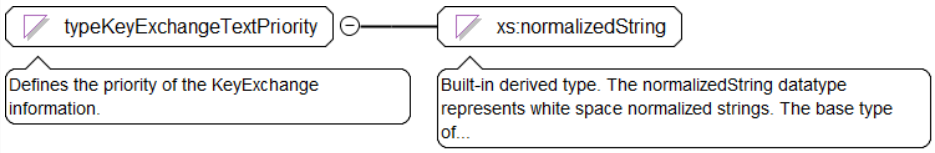
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	Represents current key state.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration keyValid — current key is valid, no user action required. enumeration keyInvalid — Key invalid, user must request key exchange. enumeration keyExchangeInProgress — Key exchange in progress, user may abort exchange or wait until it gets finished.
Used by	Element Call_KeyExchangeEvent/state
Source	<xs:simpleType name="typeKeyExchangeState"> <xs:annotation> <xs:documentation>Represents current key state.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="keyValid"> <xs:annotation> <xs:documentation>current key is valid, no user action required.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="keyInvalid"> <xs:annotation> <xs:documentation>Key invalid, user must request key exchange.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="keyExchangeInProgress"> <xs:annotation> <xs:documentation>Key exchange in progress, user may abort exchange or wait until it gets finished.</xs:documentation> </xs:annotation> </xs:enumeration> </xs:restriction> </xs:simpleType>

Simple Type typeKeyExchangeCode

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	See "Table 5.3: Status words of the commands" of the E-to-E Encryption SIM-ME Interface (Version 4.0.5) for all possible code values.
Diagram	
Type	restriction of xs:hexBinary
Facets	length 2
Used by	Element Call_KeyExchangeEvent/code
Source	<xs:simpleType name="typeKeyExchangeCode"> <xs:annotation> <xs:documentation>See "Table 5.3: Status words of the commands" of the E-to-E Encryption SIM-ME Interface (Version 4.0.5) for all possible code values.</xs:documentation>

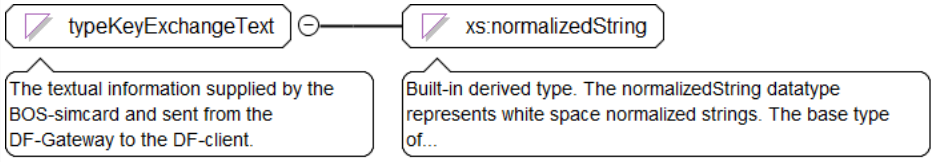
```
  </xs:annotation>
  <xs:restriction base="xs:hexBinary">
    <xs:length value="2"/>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `typeKeyExchangeTextPriority`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	`Defines the priority of the KeyExchange information.`
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `normal` enumeration `high`
Used by	Element Call_KeyExchangeEvent/priority
Source	

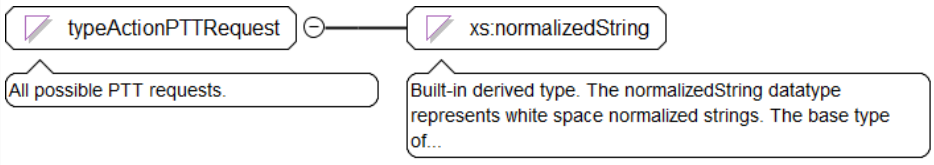
```
<xs:simpleType name="typeKeyExchangeTextPriority">
  <xs:annotation>
    <xs:documentation>Defines the priority of the KeyExchange information.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="normal"/>
    <xs:enumeration value="high"/>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `typeKeyExchangeText`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	`The textual information supplied by the BOS-simcard and sent from the DF-Gateway to the DF-client.`
Diagram	
Type	restriction of xs:normalizedString
Facets	maxLength `100`
Used by	Element Call_KeyExchangeEvent/text
Source	

```
<xs:simpleType name="typeKeyExchangeText">
  <xs:annotation>
    <xs:documentation>The textual information supplied by the BOS-simcard and sent from the DF-
Gateway to the DF-client.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:maxLength value="100"/>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `typeActionPTTRequest`

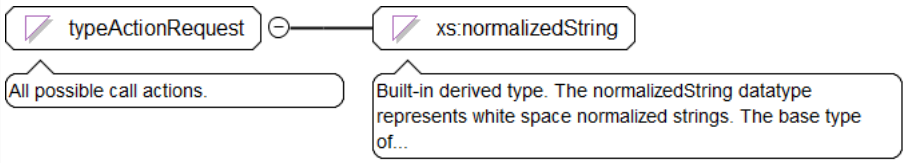
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	`All possible PTT requests.`
Diagram	

Type	restriction of xs:normalizedString		
Facets	enumeration	demandtx	This method can be used to request a speech item for a connected call.
	enumeration	ceasetx	This method is used to inform the system that the speech item is not needed any more.

Source

```
<xs:simpleType name="typeActionPTTRequest">
  <xs:annotation>
    <xs:documentation>All possible PTT requests.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="demandtx">
      <xs:annotation>
        <xs:documentation>This method can be used to request a speech item for a connected call.</
xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="ceasetx">
      <xs:annotation>
        <xs:documentation>This method is used to inform the system that the speech item is not
 needed any more.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
```

Simple Type typeActionRequest

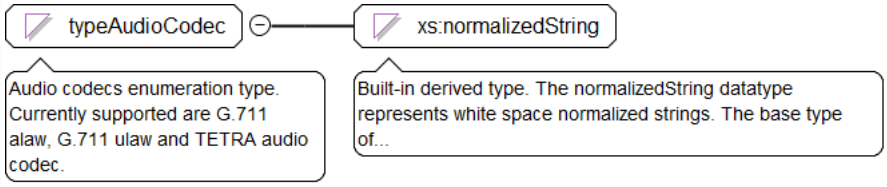
Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes		
Annotations	All possible call actions.		
Diagram			
Type	restriction of xs:normalizedString		
Facets	enumeration	setup	This method is used to initiate a new call setup. For a call setup to be successful it is required that the resources have been reserved prior this method call.
	enumeration	connect	This method is used to connect an incoming call.
	enumeration	hold	This method requests to put an individual call to hold.
	enumeration	unhold	This method is a request for resuming an individual call from hold.
	enumeration	disconnect	This method is used to disconnect a call.
	enumeration	transfer	This method is used to transfer an individual call to a new recipient.
	enumeration	releasecall	This method is used to release radio subscriber's individual call.

Source

```
<xs:simpleType name="typeActionRequest">
  <xs:annotation>
    <xs:documentation>All possible call actions.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="setup">
      <xs:annotation>
        <xs:documentation>This method is used to initiate a new call setup. For a call setup to
 be successful it is required that the resources have been reserved prior this method call.</
xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="connect">
      <xs:annotation>
        <xs:documentation>This method is used to connect an incoming call.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="hold">
      <xs:annotation>
```

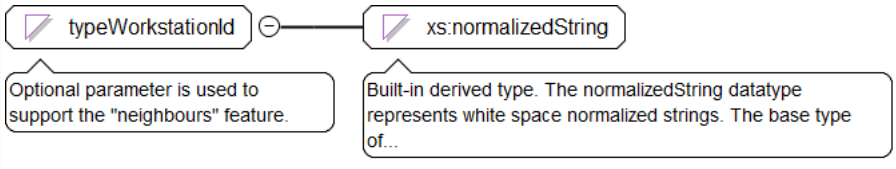
```
        <xs:documentation>This method requests to put an individual call to hold.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="unhold">
      <xs:annotation>
        <xs:documentation>This method is a request for resuming an individual call from hold.</
xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="disconnect">
      <xs:annotation>
        <xs:documentation>This method is used to disconnect a call.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="transfer">
      <xs:annotation>
        <xs:documentation>This method is used to transfer an individual call to a new recipient.</
xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="releasecall">
      <xs:annotation>
        <xs:documentation>This method is used to release radio subscriber's individual call.</
xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `typeAudioCodec`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	Audio codecs enumeration type. Currently supported are G.711 alaw, G.711 ulaw and TETRA audio codec.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `alaw` enumeration `ulaw` enumeration `tetra`
Source	

```
<xs:simpleType name="typeAudioCodec">
  <xs:annotation>
    <xs:documentation>Audio codecs enumeration type. Currently supported are G.711 alaw, G.711 ulaw
 and TETRA audio codec.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="alaw"/>
    <xs:enumeration value="ulaw"/>
    <xs:enumeration value="tetra"/>
  </xs:restriction>
</xs:simpleType>
```

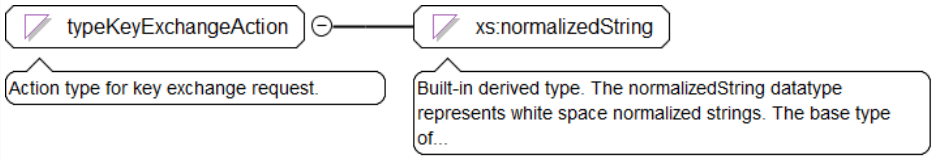
Simple Type `typeWorkstationId`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	Optional parameter is used to support the "neighbours" feature.
Diagram	
Type	xs:normalizedString
Source	

```
<xs:simpleType name="typeWorkstationId">
  <xs:annotation>
```

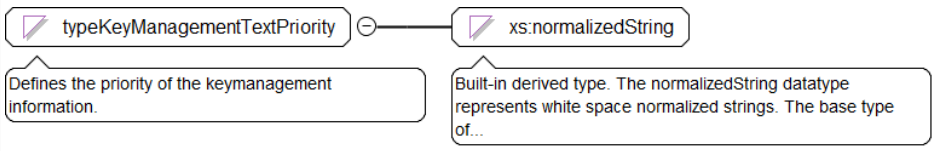
```
    <xs:documentation>Optional parameter is used to support the "neighbours" feature.</
xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
  </xs:restriction>
</xs:simpleType>
```

Simple Type `typeKeyExchangeAction`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	`Action type for key exchange request.`
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `start` enumeration `stop`

Source	`<xs:simpleType name="typeKeyExchangeAction">`
	`  <xs:annotation>`
	`  <xs:documentation>Action type for key exchange request.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="start"/>`
	`  <xs:enumeration value="stop"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `typeKeyManagementTextPriority`

Namespace	DR-GW-Interface/DR-GW-Call.CommonTypes
Annotations	`Defines the priority of the keymanagement information.`
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `normal` enumeration `high`

Source	`<xs:simpleType name="typeKeyManagementTextPriority">`
	`  <xs:annotation>`
	`  <xs:documentation>Defines the priority of the keymanagement information.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="normal"/>`
	`  <xs:enumeration value="high"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`