

Schema documentation for DR-GW.xsd

november 21, 2023

Table of Contents

Namespace: "DR-GW"	9
Schema(s)	9
Main schema DR-GW.xsd	9
Included schema DR-GW-Call.xsd	9
Included schema DR-GW-Call.select.xsd	9
Included schema DR-GW-Call.types.xsd	9
Included schema DR-GW.types.xsd	9
Included schema DR-GW-Call.call.xsd	9
Included schema DR-GW-Call.ptt.xsd	9
Included schema DR-GW-Call.keyExchange.xsd	9
Included schema DR-GW-Call.unitInEmergency.xsd	10
Included schema DR-GW-Session.xsd	10
Included schema DR-GW-Session.check.xsd	10
Included schema DR-GW-Session.login.xsd	10
Included schema DR-GW-Session.types.xsd	10
Included schema DR-GW-Session.supervise.xsd	10
Included schema DR-GW-Session.logout.xsd	10
Included schema DR-GW-Sds.xsd	10
Included schema DR-GW-Sds.send.xsd	11
Included schema DR-GW-Sds.types.xsd	11
Included schema DR-GW-Sds.receive.xsd	11
Included schema DR-GW-Status.xsd	11
Included schema DR-GW-Status.send.xsd	11
Included schema DR-GW-Status.types.xsd	11
Included schema DR-GW-Status.receive.xsd	11
Included schema DR-GW-OrganisationBlock.xsd	11
Included schema DR-GW-OrganisationBlock.get.xsd	11
Included schema DR-GW-OrganisationBlock.types.xsd	12
Included schema DR-GW-OrganisationBlock.getList.xsd	12
Included schema DR-GW-OrganisationBlock.event.xsd	12
Included schema DR-GW-Group.xsd	12
Included schema DR-GW-Group.get.xsd	12
Included schema DR-GW-Group.types.xsd	12
Included schema DR-GW-Group.getList.xsd	12
Included schema DR-GW-Group.getRadioMembers.xsd	12
Included schema DR-GW-Group.getAppMembers.xsd	13
Included schema DR-GW-Group.getCombinations.xsd	13
Included schema DR-GW-Group.track.xsd	13
Included schema DR-GW-Group.addRadioMember.xsd	13
Included schema DR-GW-Group.removeRadioMember.xsd	13
Included schema DR-GW-Group.addCombination.xsd	13
Included schema DR-GW-Group.removeCombination.xsd	13
Included schema DR-GW-Group.subscribeData.xsd	13
Included schema DR-GW-Group.event.xsd	13
Included schema DR-GW-Radio.xsd	14
Included schema DR-GW-Radio.types.xsd	14
Included schema DR-GW-Radio.get.xsd	14
Included schema DR-GW-Radio.getList.xsd	14
Included schema DR-GW-Radio.getGroups.xsd	14
Included schema DR-GW-Radio.track.xsd	14
Included schema DR-GW-Radio.changeOpta.xsd	14
Included schema DR-GW-Radio.enableDisable.xsd	14
Included schema DR-GW-Radio.event.xsd	14
Included schema DR-GW-Application.xsd	15
Included schema DR-GW-Application.get.xsd	15
Included schema DR-GW-Application.types.xsd	15
Included schema DR-GW-Application.getList.xsd	15
Included schema DR-GW-System.xsd	15
Included schema DR-GW-System.tetraStates.xsd	15
Included schema DR-GW-System.types.xsd	15
Included schema DR-GW-System.log.xsd	15
Included schema DR-GW-System.event.xsd	16
Element(s)	16

Element drgw	16
Element drgw / call	17
Element interfaceCall / select	18
Element typeRequest / requestId	18
Element typeCallSelect / target	18
Element typeSelection / level	19
Element typeSelection / target	19
Element typeAddress / subscriber	20
Element typeSubscriberAddress / ssi	20
Element typeSubscriberAddress / tsi	20
Element typeTSI / mnc	21
Element typeTSI / mcc	21
Element typeTSI / ssi	21
Element typeAddress / alias	21
Element typeAddress / msisdn	22
Element typeAddress / fssn	22
Element typeAddress / external	22
Element typeExternal / gatewayNumber	23
Element typeExternal / number	23
Element typeAddress / opta	23
Element typeAddress / cell	23
Element interfaceCall / request	24
Element typeEvent / requestId	24
Element typeEvent / result	25
Element typeResult / responseCode	25
Element typeResult / sourceSystem	25
Element typeResult / result	26
Element typeCallEvent / tetraCallId	26
Element typeCallEvent / action	26
Element typeCallEvent / attributes	26
Element typeCallAttributes / hook	27
Element typeCallAttributes / mode	27
Element typeCallAttributes / commtype	28
Element typeCallAttributes / priority	28
Element typeCallAttributes / encryption	28
Element typeCallAttributes / ambienceListen	28
Element typeCallAttributes / req2speak	29
Element typeCallAttributes / demandPriority	29
Element typeCallEvent / callingParty	29
Element typeCallEvent / calledParty	30
Element typeCallEvent / disconnectCause	31
Element typeDisconnectCause / protocol	32
Element typeDisconnectCause / code	32
Element typeDisconnectCause / text	32
Element interfaceCall / pttRequest	33
Element typeCallPTTRequest / action	33
Element typeCallPTTRequest / attributes	34
Element typeCallPTTRequest / talkingParty	34
Element typeCallPTTRequest / workstationId	35
Element interfaceCall / keyExchange	35
Element typeCallKeyExchange / action	36
Element interfaceCall / response	36
Element typeResponse / requestId	37
Element typeResponse / result	37
Element interfaceCall / selectEvent	37
Element typeCallSelectEvent / target	38
Element interfaceCall / event	38
Element interfaceCall / pttEvent	39
Element typeCallPTTEvent / tetraCallId	40
Element typeCallPTTEvent / granted	40
Element typeTxGranted / txGrant	41
Element typeTxGranted / talkingParty	42
Element typeTxGranted / encryption	42
Element typeTxGranted / txPriority	42
Element typeTxGranted / txInterrupt	43
Element typeTxGranted / txRepeat	43
Element typeTxGranted / workstationId	44
Element typeCallPTTEvent / ceased	44
Element typeCallPTTEvent / wait	44
Element interfaceCall / unitInEmergencyEvent	45
Element typeCallUnitInEmergencyEvent / group	46
Element typeCallUnitInEmergencyEvent / tetraCallId	46

Element typeCallUnitInEmergencyEvent / unitInEmg	46
Element typeCallUnitInEmergencyEvent / unitInEmgType	47
Element typeCallUnitInEmergencyEvent / emgInfo	47
Element typeCallUnitInEmergencyEvent / tstamp	48
Element interfaceCall / keyExchangeEvent	48
Element typeCallKeyExchangeEvent / state	49
Element typeCallKeyExchangeEvent / code	49
Element typeCallKeyExchangeEvent / priority	49
Element typeCallKeyExchangeEvent / interaction	50
Element typeCallKeyExchangeEvent / text	50
Element typeCallKeyExchangeEvent / tone	50
Element drgw / session	51
Element interfaceSession / login	51
Element typeSessionLogin / clientid	52
Element typeSessionLogin / supervise	52
Element typeSessionLogin / version	52
Element interfaceSession / logout	53
Element interfaceSession / supervise	53
Element interfaceSession / check	53
Element typeSessionCheck / clientid	54
Element interfaceSession / response	54
Element interfaceSession / loginEvent	55
Element typeSessionLoginEvent / issi	55
Element interfaceSession / superviseEvent	55
Element drgw / sds	56
Element interfaceSds / send	56
Element typeSdsSend / sds	57
Element typeSds / protocol	58
Element typeSds / sdsType	58
Element typeSds / msgRef	58
Element typeSds / report	58
Element typeSds / sdsdata	59
Element typeSdsData / data	59
Element typeSdsData / hexdata	59
Element typeSdsData / hexdatalength	60
Element typeSds / source	60
Element typeSds / target	60
Element typeSds / forward	61
Element typeSds / validity	62
Element typeSds / tstamp	62
Element typeSds / encryption	63
Element typeSds / e2eegroup	63
Element interfaceSds / sendReport	63
Element typeSdsSendReport / target	64
Element typeSdsSendReport / msgRef	64
Element typeSdsSendReport / deliveryStatus	65
Element interfaceSds / response	65
Element interfaceSds / sendEvent	65
Element typeSdsSendEvent / msgRef	66
Element typeSdsSendEvent / sds	66
Element interfaceSds / receiveEvent	67
Element typeSdsReceiveEvent / sds	68
Element interfaceSds / reportEvent	69
Element typeSdsReportEvent / source	70
Element typeSdsReportEvent / target	70
Element typeSdsReportEvent / msgRef	71
Element typeSdsReportEvent / deliveryStatus	71
Element typeSdsReportEvent / tstamp	72
Element drgw / status	72
Element interfaceStatus / send	72
Element typeStatusSend / status	73
Element typeStatus / value	73
Element typeStatus / hexValue	73
Element typeStatus / source	74
Element typeStatus / target	74
Element typeStatus / tstamp	75
Element interfaceStatus / response	75
Element interfaceStatus / sendEvent	76
Element typeStatusSendEvent / status	76
Element interfaceStatus / receiveEvent	77
Element typeStatusReceiveEvent / status	77
Element drgw / org	78

Element interfaceOrg / get	79
Element typeOrgGet / orgblockId	79
Element typeOrganisationBlockId / orgblockId	79
Element typeOrganisationBlockIdNormal / id1	80
Element typeOrganisationBlockIdNormal / id2	80
Element typeOrganisationBlockIdNormal / id3	81
Element typeOrganisationBlockIdNormal / id4	81
Element typeOrganisationBlockIdNormal / id5	81
Element typeOrganisationBlockIdNormal / id6	81
Element typeOrganisationBlockId / orgblockIdSimple	82
Element interfaceOrg / getList	82
Element typeOrgGetList / orgblockId	82
Element interfaceOrg / response	83
Element interfaceOrg / getEvent	83
Element typeOrgGetEvent / orgblock	84
Element typeOrganisationBlock / orgblockId	84
Element typeOrganisationBlock / alias	84
Element interfaceOrg / getListEvent	85
Element typeOrgGetListEvent / orgblock	85
Element typeOrgGetListEvent / listEnd	85
Element interfaceOrg / event	86
Element typeOrgEvent / orgblock	86
Element typeOrgEvent / delete	87
Element drgw / group	87
Element interfaceGroup / get	89
Element typeGroupGet / group	90
Element interfaceGroup / getList	90
Element typeGroupGetList / orgblockId	90
Element interfaceGroup / getRadioMembers	91
Element typeGroupGetRadioMembers / group	91
Element interfaceGroup / getAppMembers	92
Element typeGroupGetAppMembers / group	92
Element interfaceGroup / getCombinations	92
Element typeGroupGetCombinations / group	93
Element interfaceGroup / track	93
Element typeGroupTrack / group	94
Element typeGroupTrack / mask	94
Element typeGroupTrack / stop	95
Element interfaceGroup / addRadioMember	95
Element typeGroupAddRadioMember / radio	95
Element typeGroupAddRadioMember / group	96
Element typeGroupAddRadioMember / membership	96
Element interfaceGroup / removeRadioMember	96
Element typeGroupRemoveRadioMember / radio	97
Element typeGroupRemoveRadioMember / group	97
Element interfaceGroup / addCombination	98
Element typeGroupAddCombination / group	98
Element typeGroupAddCombination / baseGroup	99
Element typeGroupAddCombination / force	99
Element interfaceGroup / removeCombination	99
Element typeGroupRemoveCombination / group	100
Element typeGroupRemoveCombination / baseGroup	100
Element interfaceGroup / subscribeData	101
Element typeGroupSubscribeData / group	101
Element typeGroupDataSubscription / addr	102
Element typeGroupDataSubscription / useSDS	102
Element typeGroupDataSubscription / useStatus	102
Element interfaceGroup / response	102
Element interfaceGroup / getEvent	103
Element typeGroupGetEvent / group	103
Element typeGroup / addr	104
Element typeGroup / alias	104
Element typeGroup / orgblockId	104
Element interfaceGroup / getListEvent	105
Element typeGroupGetListEvent / group	105
Element typeGroupGetListEvent / listEnd	105
Element interfaceGroup / getRadioMembersEvent	106
Element typeGroupGetRadioMembersEvent / group	106
Element typeGroupGetRadioMembersEvent / radio	107
Element typeGroupGetRadioMembersEvent / listEnd	107
Element interfaceGroup / getAppMembersEvent	107
Element typeGroupGetAppMembersEvent / app	108

Element typeGroupGetAppMembersEvent / listEnd	108
Element interfaceGroup / trackSubscriptionEvent	108
Element typeGroupTrackSubscriptionEvent / group	109
Element typeGroupTrackSubscriptionEvent / mask	109
Element typeGroupTrackSubscriptionEvent / stop	110
Element interfaceGroup / radioMemberEvent	110
Element typeGroupRadioMemberEvent / group	110
Element typeGroupRadioMemberEvent / radio	111
Element typeGroupRadioMemberEvent / delete	111
Element interfaceGroup / appMemberEvent	111
Element typeGroupAppMemberEvent / group	112
Element typeGroupAppMemberEvent / app	112
Element typeGroupAppMemberEvent / delete	113
Element interfaceGroup / combinationEvent	113
Element typeGroupCombinationEvent / group	114
Element typeGroupCombinationEvent / baseGroup	114
Element typeGroupCombinationEvent / constitGroup	114
Element interfaceGroup / addRadioMemberEvent	115
Element typeGroupAddRadioMemberEvent / radio	115
Element typeGroupAddRadioMemberEvent / group	116
Element interfaceGroup / removeRadioMemberEvent	116
Element typeGroupRemoveRadioMemberEvent / radio	117
Element typeGroupRemoveRadioMemberEvent / group	117
Element interfaceGroup / addCombinationEvent	117
Element typeGroupAddCombinationEvent / group	118
Element typeGroupAddCombinationEvent / baseGroup	118
Element interfaceGroup / removeCombinationEvent	118
Element typeGroupRemoveCombinationEvent / group	119
Element typeGroupRemoveCombinationEvent / baseGroup	119
Element interfaceGroup / subscribeDataEvent	120
Element typeGroupSubscribeDataEvent / group	120
Element interfaceGroup / event	121
Element typeGroupEvent / group	121
Element typeGroupEvent / delete	122
Element drgw / radio	122
Element interfaceRadio / get	123
Element typeRadioGet / radio	123
Element interfaceRadio / getList	124
Element typeRadioGetList / orgblockId	124
Element interfaceRadio / getGroups	125
Element typeRadioGetGroups / radio	125
Element typeRadio / issi	125
Element typeRadio / alias	126
Element typeRadio / orgblockId	126
Element typeRadio / opta	126
Element typeLastKnownOPTA / tstamp	127
Element typeLastKnownOPTA / opta	127
Element interfaceRadio / track	127
Element typeRadioTrack / radio	128
Element typeRadioTrack / stop	128
Element interfaceRadio / changeOpta	128
Element typeRadioChangeOpta / radio	129
Element typeRadioChangeOpta / opta	129
Element interfaceRadio / enable	130
Element typeRadioEnable / radio	130
Element typeRadioEnable / reason	131
Element typeRadioEnable / enable	131
Element interfaceRadio / disable	131
Element typeRadioDisable / radio	132
Element typeRadioDisable / reason	132
Element typeRadioDisable / enable	132
Element interfaceRadio / response	132
Element interfaceRadio / getEvent	133
Element typeRadioGetEvent / radio	133
Element interfaceRadio / getListEvent	134
Element typeRadioGetListEvent / radio	134
Element typeRadioGetListEvent / listEnd	135
Element interfaceRadio / getGroupsEvent	135
Element typeRadioGetGroupsEvent / radio	136
Element typeRadioGetGroupsEvent / group	136
Element typeRadioGroupSelection / group	136
Element typeRadioGroupSelection / level	137

Element typeRadioGetGroupsEvent / listEnd	137
Element interfaceRadio / trackSubscriptionEvent	137
Element typeRadioTrackSubscriptionEvent / radio	138
Element typeRadioTrackSubscriptionEvent / stop	138
Element interfaceRadio / trackEvent	139
Element typeRadioTrackEvent / trackingData	139
Element typeRadioTrackingData / radio	140
Element typeRadioTrackingData / registered	141
Element typeRadioTrackingData / exchangeId	141
Element typeRadioTrackingData / locationArea	141
Element typeRadioTrackingData / lastActive	141
Element typeRadioTrackingData / scanningOn	142
Element typeRadioTrackingData / status	142
Element typeStatusIndicator / value	142
Element typeStatusIndicator / time	142
Element typeRadioTrackingData / callType	143
Element typeRadioTrackingData / callParty	143
Element typeRadioTrackingData / dmoState	143
Element typeRadioTrackingData / emergency	143
Element interfaceRadio / groupsEvent	144
Element typeRadioGroupsEvent / radio	144
Element typeRadioGroupsEvent / group	145
Element typeRadioGroupsEvent / deletedGroup	145
Element interfaceRadio / changeOptaEvent	145
Element typeRadioChangeOptaEvent / radio	146
Element typeRadioChangeOptaEvent / opta	146
Element interfaceRadio / enableDisableEvent	146
Element typeRadioEnableDisableEvent / radio	147
Element typeRadioEnableDisableEvent / reason	147
Element typeRadioEnableDisableEvent / enabled	148
Element typeRadioEnableDisableEvent / overTheAir	148
Element interfaceRadio / event	148
Element typeRadioEvent / radio	149
Element typeRadioEvent / delete	149
Element drgw / app	149
Element interfaceApp / get	150
Element typeAppGet / app	150
Element interfaceApp / getList	151
Element typeAppGetList / orgblockId	151
Element interfaceApp / response	152
Element interfaceApp / getEvent	152
Element typeAppGetEvent / app	153
Element typeApplication / addr	153
Element typeApplication / alias	153
Element typeApplication / orgblockId	154
Element interfaceApp / getListEvent	154
Element typeAppGetListEvent / app	154
Element typeAppGetListEvent / listEnd	155
Element drgw / system	155
Element interfaceSystem / tetraStatesEvent	156
Element typeSystemTetraStatesEvent / tcsState	156
Element typeSystemTetraStatesEvent / dxtState	157
Element typeSystemTetraStatesEvent / cddconnectionState	157
Element typeSystemTetraStatesEvent / cddserverState	157
Element interfaceSystem / logEvent	157
Element typeSystemLogEvent / value	158
Element typeSystemLogEvent / text	158
Element interfaceSystem / event	159
Element typeSystemEvent / value	159
Element typeSystemEvent / text	159
Element typeCallRequest / action	160
Element typeCallRequest / attributes	160
Element typeCallRequest / callingParty	161
Element typeCallRequest / calledParty	161
Element typeCallRequest / workstationId	162
Element check	162
Element typeSessionLogoutEvent / reason	163
Element typeSdsValidity / value	163
Element typeGroupGetCombinationsEvent / group	163
Element typeGroupGetCombinationsEvent / baseGroup	164
Element typeGroupGetCombinationsEvent / constitGroup	164
Complex Type(s)	165

Complex Type interfaceCall	165
Complex Type typeCallSelect	165
Complex Type typeRequest	166
Complex Type typeSelection	166
Complex Type typeAddress	167
Complex Type typeSubscriberAddress	167
Complex Type typeTSI	168
Complex Type typeExternal	169
Complex Type typeCallEvent	169
Complex Type typeEvent	170
Complex Type typeResult	170
Complex Type typeCallAttributes	171
Complex Type typeDisconnectCause	171
Complex Type typeCallPTTRequest	172
Complex Type typeCallKeyExchange	173
Complex Type typeResponse	173
Complex Type typeCallSelectEvent	173
Complex Type typeCallPTTEvent	174
Complex Type typeTxGranted	175
Complex Type typeEmpty	176
Complex Type typeCallUnitInEmergencyEvent	176
Complex Type typeCallKeyExchangeEvent	177
Complex Type interfaceSession	178
Complex Type typeSessionLogin	179
Complex Type typeSessionLogout	180
Complex Type typeSessionSupervise	180
Complex Type typeSessionCheck	181
Complex Type typeSessionLoginEvent	181
Complex Type typeSessionSuperviseEvent	182
Complex Type interfaceSds	182
Complex Type typeSdsSend	183
Complex Type typeSds	183
Complex Type typeSdsData	184
Complex Type typeSdsSendReport	185
Complex Type typeSdsSendEvent	185
Complex Type typeSdsReceiveEvent	186
Complex Type typeSdsReportEvent	187
Complex Type interfaceStatus	187
Complex Type typeStatusSend	188
Complex Type typeStatus	188
Complex Type typeStatusSendEvent	189
Complex Type typeStatusReceiveEvent	189
Complex Type interfaceOrg	190
Complex Type typeOrgGet	191
Complex Type typeOrganisationBlockId	191
Complex Type typeOrganisationBlockIdNormal	191
Complex Type typeOrgGetList	192
Complex Type typeOrgGetEvent	192
Complex Type typeOrganisationBlock	193
Complex Type typeOrgGetListEvent	193
Complex Type typeOrgEvent	194
Complex Type interfaceGroup	194
Complex Type typeGroupGet	196
Complex Type typeGroupGetList	197
Complex Type typeGroupGetRadioMembers	197
Complex Type typeGroupGetAppMembers	198
Complex Type typeGroupGetCombinations	198
Complex Type typeGroupTrack	199
Complex Type typeGroupAddRadioMember	199
Complex Type typeGroupRemoveRadioMember	200
Complex Type typeGroupAddCombination	200
Complex Type typeGroupRemoveCombination	201
Complex Type typeGroupSubscribeData	202
Complex Type typeGroupDataSubscription	202
Complex Type typeGroupGetEvent	202
Complex Type typeGroup	203
Complex Type typeGroupGetListEvent	203
Complex Type typeGroupGetRadioMembersEvent	204
Complex Type typeGroupGetAppMembersEvent	205
Complex Type typeGroupTrackSubscriptionEvent	205
Complex Type typeGroupRadioMemberEvent	206
Complex Type typeGroupAppMemberEvent	206

Complex Type typeGroupCombinationEvent	207
Complex Type typeGroupAddRadioMemberEvent	208
Complex Type typeGroupRemoveRadioMemberEvent	208
Complex Type typeGroupAddCombinationEvent	209
Complex Type typeGroupRemoveCombinationEvent	209
Complex Type typeGroupSubscribeDataEvent	210
Complex Type typeGroupEvent	211
Complex Type interfaceRadio	211
Complex Type typeRadioGet	213
Complex Type typeRadioGetList	213
Complex Type typeRadioGetGroups	213
Complex Type typeRadio	214
Complex Type typeLastKnownOPTA	214
Complex Type typeRadioTrack	215
Complex Type typeRadioChangeOpta	215
Complex Type typeRadioEnable	216
Complex Type typeRadioDisable	216
Complex Type typeRadioGetEvent	217
Complex Type typeRadioGetListEvent	218
Complex Type typeRadioGetGroupsEvent	218
Complex Type typeRadioGroupSelection	219
Complex Type typeRadioTrackSubscriptionEvent	219
Complex Type typeRadioTrackEvent	220
Complex Type typeRadioTrackingData	220
Complex Type typeStatusIndicator	221
Complex Type typeRadioGroupsEvent	221
Complex Type typeRadioChangeOptaEvent	222
Complex Type typeRadioEnableDisableEvent	222
Complex Type typeRadioEvent	223
Complex Type interfaceApp	223
Complex Type typeAppGet	224
Complex Type typeAppGetList	224
Complex Type typeAppGetEvent	225
Complex Type typeApplication	226
Complex Type typeAppGetListEvent	226
Complex Type interfaceSystem	226
Complex Type typeSystemTetraStatesEvent	227
Complex Type typeSystemLogEvent	228
Complex Type typeSystemEvent	228
Complex Type typeCallRequest	229
Complex Type typeSessionLogoutEvent	230
Complex Type typeSdsValidity	230
Complex Type typeGroupGetCombinationsEvent	230
Simple Type(s)	231
Simple Type typeSelectionLevel	231
Simple Type typeDialString	232
Simple Type typeOpta	232
Simple Type typeResponseCode	233
Simple Type typeSourceSystem	233
Simple Type typeActionEvent	233
Simple Type typeCallMode	234
Simple Type typeCallType	235
Simple Type typeTxDemandPriority	235
Simple Type typeActionPTTRequest	236
Simple Type typeWorkstationId	236
Simple Type typeKeyExchangeAction	236
Simple Type typeTxGrant	237
Simple Type typeTxPriority	237
Simple Type typeUnitInEmergencyType	238
Simple Type typeEmergencyInfo	238
Simple Type typeKeyExchangeState	239
Simple Type typeKeyExchangeCode	239
Simple Type typeKeyExchangeTextPriority	240
Simple Type typeKeyExchangeText	240
Simple Type typeSuperviseTimeout	240
Simple Type typeSdsType	241
Simple Type typeReport	242
Simple Type typeOrganisationBlockIdSimple	242
Simple Type typeGroupTrackingMask	243
Simple Type typeMembershipType	243
Simple Type typeGroupSelectionLevel	243
Simple Type typeSystemElementState	244

Simple Type typeActionRequest	244
Simple Type typeKeyManagementTextPriority	245
Simple Type typeAddressingStyle	246
Simple Type typeGroupTrackingMaskValues	246
Attribute(s)	247
Attribute drgw / @version	247

Namespace: "DR-GW"

Schema(s)

Main schema DR-GW.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Call.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Call.select.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Call.types.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW.types.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Call.call.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Call.ppt.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Call.keyExchange.xsd

Namespace	DR-GW
-----------	-------

Properties	attribute form default: qualified element form default: qualified
------------	--

Included schema DR-GW-Call.unitInEmergency.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Session.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Session.check.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Session.login.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Session.types.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Session.supervise.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Session.logout.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Sds.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Sds.send.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Sds.types.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Sds.receive.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Status.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Status.send.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Status.types.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Status.receive.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-OrganisationBlock.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-OrganisationBlock.get.xsd

Namespace	DR-GW

Properties	attribute form default: qualified element form default: qualified
------------	--

Included schema DR-GW-OrganisationBlock.types.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-OrganisationBlock.getList.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-OrganisationBlock.event.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.get.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.types.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.getList.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.getRadioMembers.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.getAppMembers.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.getCombinations.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.track.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.addRadioMember.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.removeRadioMember.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.addCombination.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.removeCombination.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.subscribeData.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.event.xsd

Namespace	DR-GW
Properties	attribute form default: qualified

	element form default: qualified
--	---------------------------------

Included schema DR-GW-Radio.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Radio.types.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Radio.get.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Radio.getList.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Radio.getGroups.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Radio.track.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Radio.changeOpta.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Radio.enableDisable.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Radio.event.xsd

Namespace	DR-GW

Properties	attribute form default: qualified element form default: qualified
------------	--

Included schema DR-GW-Application.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Application.get.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Application.types.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Application.getList.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-System.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-System.tetraStates.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-System.types.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-System.log.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-System.event.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Element(s)**Element drgw**

Namespace	DR-GW
Annotations	Root DR-GW element.
Diagram	<pre> classDiagram class drgw { @version : Fixed 2.0 call session sds status org group radio app system } </pre>
Properties	content: complex
Model	call session sds status org group radio app system
Children	app, call, group, org, radio, sds, session, status, system
Instance	<pre> <drgw version="2.0" xmlns="DR-GW"> <call>{1,1}</call> <session>{1,1}</session> <sds>{1,1}</sds> <status>{1,1}</status> <org>{1,1}</org> <group>{1,1}</group> <radio>{1,1}</radio> <app>{1,1}</app> <system>{1,1}</system> </pre>

	</drgw>				
Attributes	QName	Type	Fixed	Use	
Source	<pre><xs:element name="drgw"> <xs:annotation> <xs:documentation>Root DR-GW element.</xs:documentation> </xs:annotation> <xs:complexType> <xs:choice> <xs:element name="call" type="interfaceCall"/> <xs:element name="session" type="interfaceSession"/> <xs:element name="sds" type="interfaceSds"/> <xs:element name="status" type="interfaceStatus"/> <xs:element name="org" type="interfaceOrg"/> <xs:element name="group" type="interfaceGroup"/> <xs:element name="radio" type="interfaceRadio"/> <xs:element name="app" type="interfaceApp"/> <xs:element name="system" type="interfaceSystem"/> </xs:choice> <xs:attribute name="version" fixed="2.0" use="required"/> </xs:complexType> </xs:element></pre>				

Element drgw / call

Namespace	DR-GW
Diagram	<p>The diagram illustrates the UML class 'call' which has a dependency on the interface 'interfaceCall'. The 'interfaceCall' interface defines a sequence of events: select, request, pttRequest, keyExchange, response, selectEvent, event, pttEvent, unitInEmergencyEvent, and keyExchangeEvent. A note at the bottom states: 'DR-GW-Call. Use for call control/ call monitoring. This is the only element, that can be used via both SIP/SOAP. When...'.</p>
Type	interfaceCall
Properties	content: complex
Model	select request pttRequest keyExchange response selectEvent event pttEvent unitInEmergencyEvent keyExchangeEvent
Children	event, keyExchange, keyExchangeEvent, pttEvent, pttRequest, request, response, select, selectEvent, unitInEmergencyEvent
Instance	<pre><call xmlns="DR-GW"> <select>{1,1}</select> <request>{1,1}</request> <pttRequest>{1,1}</pttRequest> <keyExchange>{1,1}</keyExchange> <response>{1,1}</response> <selectEvent>{1,1}</selectEvent></pre>

	<pre> <event>{1,1}</event> <pttEvent>{1,1}</pttEvent> <unitInEmergencyEvent>{1,1}</unitInEmergencyEvent> <keyExchangeEvent>{1,1}</keyExchangeEvent> </call> </pre>
Source	<xss:element name="call" type="interfaceCall" />

Element interfaceCall / select

Namespace	DR-GW
Diagram	<p>The method reserves speech line for the targets of selection operation, sets the selection level of the targets of...</p>
Type	typeCallSelect
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeCallSelect
Properties	content: complex
Model	requestId , target
Children	requestId, target
Instance	<pre> <select xmlns="DR-GW"> <requestId>{1,1}</requestId> <target>{1,1}</target> </select> </pre>
Source	<xss:element name="select" type="typeCallSelect" />

Element typeRequest / requestId

Namespace	DR-GW
Diagram	<p>Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...</p>
Type	xs:unsignedLong
Properties	content: simple
Source	<xss:element name="requestId" type="xs:unsignedLong" />

Element typeCallSelect / target

Namespace	DR-GW
Diagram	

Type	typeSelection
Properties	content: complex
Model	level , target
Children	level, target
Instance	<target xmlns="DR-GW"> <level>{1,1}</level> <target>{1,1}</target> </target>
Source	<xss:element name="target" type="typeSelection"/>

Element typeSelection / level

Namespace	DR-GW															
Diagram	<pre> graph LR level((level)) -- "-" --> typeSelectionLevel((typeSelectionLevel)) subgraph Callout [Defines how the target is monitored.] direction TB L1["Defines how the target is monitored."] end </pre>															
Type	typeSelectionLevel															
Properties	content: simple															
Facets	<table> <tr> <td>enumeration</td> <td>no</td> <td>No selection. Used to remove selection.</td> </tr> <tr> <td>enumeration</td> <td>event</td> <td>Event monitoring.</td> </tr> <tr> <td>enumeration</td> <td>audio</td> <td>Audio monitoring.</td> </tr> <tr> <td>enumeration</td> <td>use</td> <td>Selection level use.</td> </tr> <tr> <td>enumeration</td> <td>a_use</td> <td>Selection level active use.</td> </tr> </table>	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.
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	<xss:element name="level" type="typeSelectionLevel"/>															

Element typeSelection / target

Namespace	DR-GW
Diagram	<pre> graph TD target((target)) -- "-" --> typeAddress((typeAddress)) subgraph Callout [Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).] direction TB L1["Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA)."] L2["subscriber"] L2["alias"] L2["msisdn"] L2["fssn"] L2["external"] L2["opta"] L2["cell"] end </pre>
Type	typeAddress
Properties	content: complex
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber

Instance	<pre><target xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </target></pre>
Source	<pre><xss:element name="target" type="typeAddress" /></pre>

Element typeAddress / subscriber

Namespace	DR-GW				
Diagram	<pre> classDiagram class subscriber class typeSubscriberAddress { <<ssi>> <<tsi>> } subscriber --> typeSubscriberAddress </pre>				
Type	typeSubscriberAddress				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	ssi tsi				
Children	ssi, tsi				
Instance	<pre><subscriber xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </subscriber></pre>				
Source	<pre><xss:element name="subscriber" type="typeSubscriberAddress" minOccurs="0" /></pre>				

Element typeSubscriberAddress / ssi

Namespace	DR-GW		
Diagram	<pre> classDiagram class ssi class xsUnsignedLong { <<xs:unsignedLong>> } ssi --> xsUnsignedLong </pre> <p>Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...</p>		
Type	xs:unsignedLong		
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple
content:	simple		
Source	<pre><xss:element name="ssi" type="xs:unsignedLong" /></pre>		

Element typeSubscriberAddress / tsi

Namespace	DR-GW
Diagram	<pre> classDiagram class tsi class typeTSI { <<mnc>> <<mcc>> <<ssi>> } tsi --> typeTSI </pre> <p>Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC).</p>

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

Element typeTSI / mnc

Namespace	DR-GW
Diagram	<pre> graph LR mnc[mnc] --> xs["xs:unsignedShort"] </pre> <p>Built-in derived type. The unsignedShort datatype is derived from unsignedInt by setting the value of maxInclusive to...</p>
Type	xs:unsignedShort
Properties	content: simple
Source	<xss:element name="mnc" type="xs:unsignedShort" />

Element typeTSI / mcc

Namespace	DR-GW
Diagram	<pre> graph LR mcc[mcc] --> xs["xs:unsignedShort"] </pre> <p>Built-in derived type. The unsignedShort datatype is derived from unsignedInt by setting the value of maxInclusive to...</p>
Type	xs:unsignedShort
Properties	content: simple
Source	<xss:element name="mcc" type="xs:unsignedShort" />

Element typeTSI / ssi

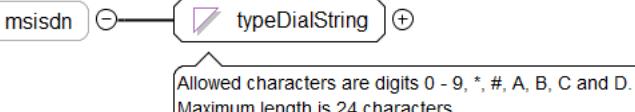
Namespace	DR-GW
Diagram	<pre> graph LR ssi[ssi] --> xs["xs:unsignedLong"] </pre> <p>Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...</p>
Type	xs:unsignedLong
Properties	content: simple
Source	<xss:element name="ssi" type="xs:unsignedLong" />

Element typeAddress / alias

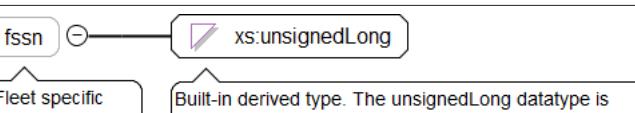
Namespace	DR-GW
Diagram	<pre> graph LR alias[alias] --> xs["xs:normalizedString"] </pre> <p>Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...</p>

Type	xs:normalizedString
Properties	content: simple minOccurs: 0
Source	<xs:element name="alias" type="xs:normalizedString" minOccurs="0" />

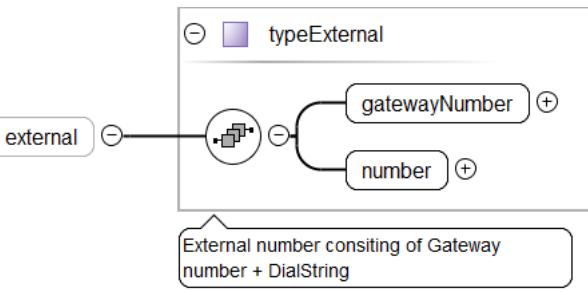
Element typeAddress / msisdn

Namespace	DR-GW
Diagram	
Type	typeDialString
Properties	content: simple minOccurs: 0
Facets	maxLength 24
Source	<xs:element name="msisdn" type="typeDialString" minOccurs="0" />

Element typeAddress / fssn

Namespace	DR-GW
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 typeAddress / external

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

Instance	<pre><external xmlns="DR-GW"> <gatewayNumber>{1,1}</gatewayNumber> <number>{1,1}</number> </external></pre>
Source	<pre><xss:element name="external" type="typeExternal" minOccurs="0" /></pre>

Element typeExternal / gatewayNumber

Namespace	DR-GW
Diagram	<pre> classDiagram class gatewayNumber { <<xs:unsignedLong>> } gatewayNumber < -- xs:unsignedLong </pre> <p>Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...</p>
Type	xs:unsignedLong
Properties	content: simple
Source	<pre><xss:element name="gatewayNumber" type="xs:unsignedLong" /></pre>

Element typeExternal / number

Namespace	DR-GW
Diagram	<pre> classDiagram class number { <<typeDialString>> } number < -- typeDialString </pre> <p>Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.</p>
Type	typeDialString
Properties	content: simple
Facets	maxLength 24
Source	<pre><xss:element name="number" type="typeDialString" /></pre>

Element typeAddress / opta

Namespace	DR-GW
Diagram	<pre> classDiagram class opta { <<typeOpta>> } opta < -- typeOpta </pre> <p>OPTA string. Maximum length is 24 characters.</p>
Type	typeOpta
Properties	content: simple minOccurs: 0
Facets	maxLength 24
Source	<pre><xss:element name="opta" type="typeOpta" minOccurs="0" /></pre>

Element typeAddress / cell

Namespace	DR-GW
Diagram	<pre> classDiagram class cell { <<xs:short>> } cell < -- xs:short </pre> <p>Built-in derived type. The short datatype is derived from int by setting the value of maxInclusive to be 32767 and...</p>
Type	xs:short
Properties	content: simple

	minOccurs:	0
Source	<xs:element name="cell" type="xs:short" minOccurs="0"/>	

Element interfaceCall / request

Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeCallEvent typeCallEvent "0..1" -- "0..1" requestId : xs:short typeCallEvent "0..1" -- "0..1" result : xs:string typeCallEvent "0..1" -- "0..1" tetraCallId : xs:unsignedLong typeCallEvent "0..1" -- "0..1" action : xs:string typeCallEvent "0..1" -- "0..1" attributes : xs:string typeCallEvent "0..1" -- "0..1" callingParty : xs:string typeCallEvent "0..1" -- "0..1" calledParty : xs:string typeCallEvent "0..1" -- "0..1" disconnectCause : xs:string typeCallEvent "0..1" -- "0..1" request : xs:string </pre>
Type	typeCallEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent <ul style="list-style-type: none"> • typeCallEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , tetraCallId{0,1} , action , attributes{0,1} , callingParty{0,1} , calledParty{0,1} , disconnectCause{0,1}
Children	action, attributes, calledParty, callingParty, disconnectCause, requestId, result, tetraCallId
Instance	<pre> <request xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</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> </request> </pre>
Source	<xs:element name="request" type="typeCallEvent" />

Element typeEvent / requestId

Namespace	DR-GW				
Diagram	<pre> requestId <--> xs:unsignedLong </pre> <p>Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...</p>				
Type	xs:unsignedLong				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				

Source	<code><xss:element name="requestId" type="xs:unsignedLong" minOccurs="0" /></code>
--------	--

Element typeEvent / result

Namespace	DR-GW				
Diagram	<pre> classDiagram class typeResult { responseCode sourceSystem result } class result typeResult < -- result </pre> <p>Common result values used in every response and optional specific subsystem result codes.</p>				
Type	typeResult				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	responseCode , sourceSystem{0,1} , result{0,1}				
Children	responseCode, result, sourceSystem				
Instance	<pre> <result xmlns="DR-GW"> <responseCode>{1,1}</responseCode> <sourceSystem>{0,1}</sourceSystem> <result>{0,1}</result> </result> </pre>				
Source	<code><xss:element name="result" type="typeResult" minOccurs="0" /></code>				

Element typeResult / responseCode

Namespace	DR-GW												
Diagram	<pre> classDiagram class responseCode { typeResponseCode } </pre>												
Type	typeResponseCode												
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple										
content:	simple												
Facets	<table border="1"> <tr> <td>enumeration</td> <td>success</td> </tr> <tr> <td>enumeration</td> <td>final_response_pending</td> </tr> <tr> <td>enumeration</td> <td>error</td> </tr> <tr> <td>enumeration</td> <td>not_authorized_error</td> </tr> <tr> <td>enumeration</td> <td>temporary_failure</td> </tr> <tr> <td>enumeration</td> <td>subscription_failed</td> </tr> </table>	enumeration	success	enumeration	final_response_pending	enumeration	error	enumeration	not_authorized_error	enumeration	temporary_failure	enumeration	subscription_failed
enumeration	success												
enumeration	final_response_pending												
enumeration	error												
enumeration	not_authorized_error												
enumeration	temporary_failure												
enumeration	subscription_failed												
Source	<code><xss:element name="responseCode" type="typeResponseCode" /></code>												

Element typeResult / sourceSystem

Namespace	DR-GW				
Diagram	<pre> classDiagram class sourceSystem { typeSourceSystem } </pre>				
Type	typeSourceSystem				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Facets	<table border="1"> <tr> <td>enumeration</td> <td>DR-GW</td> </tr> </table>	enumeration	DR-GW		
enumeration	DR-GW				

	enumeration	TCS-API
	enumeration	TETRA
Source	<xss:element name="sourceSystem" type="typeSourceSystem" minOccurs="0" />	

Element typeResult / result

Namespace	DR-GW				
Diagram	<p>Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...</p>				
Type	xs:unsignedLong				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<xss:element name="result" type="xs:unsignedLong" minOccurs="0" />				

Element typeCallEvent / tetraCallId

Namespace	DR-GW				
Diagram	<p>Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...</p>				
Type	xs:unsignedLong				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<xss:element name="tetraCallId" type="xs:unsignedLong" minOccurs="0" />				

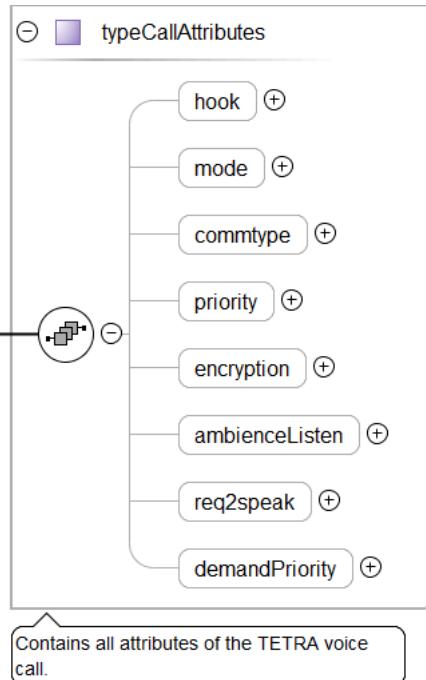
Element typeCallEvent / action

Namespace	DR-GW																		
Diagram	<p>All possible call actions.</p>																		
Type	typeActionEvent																		
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple																
content:	simple																		
Facets	<table border="1"> <tr> <td>enumeration</td> <td>incoming</td> <td>This event fired when there is an incoming call. This is the first indication of a new incoming call.</td> </tr> <tr> <td>enumeration</td> <td>connected</td> <td>This event is used to inform that call has been connected and call setup is finished.</td> </tr> <tr> <td>enumeration</td> <td>held</td> <td>This event is used to inform TCS Client that individual call was put to hold.</td> </tr> <tr> <td>enumeration</td> <td>resumed</td> <td>This event is used to inform that individual call has been taken from hold.</td> </tr> <tr> <td>enumeration</td> <td>disconnected</td> <td>This event is used to inform that the call was disconnected.</td> </tr> <tr> <td>enumeration</td> <td>transferred</td> <td>This event is a response to transfer method call and indicates the result of the request.</td> </tr> </table>	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.
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	<xss:element name="action" type="typeActionEvent" />																		

Element typeCallEvent / attributes

Namespace	DR-GW
-----------	-------

Diagram



Type	<code>typeCallAttributes</code>				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	<code>hook{0,1}</code> , <code>mode{0,1}</code> , <code>commtype{0,1}</code> , <code>priority{0,1}</code> , <code>encryption{0,1}</code> , <code>ambienceListen{0,1}</code> , <code>req2speak{0,1}</code> , <code>demandPriority{0,1}</code>				
Children	ambienceListen, commtype, demandPriority, encryption, hook, mode, priority, req2speak				
Instance	<pre><attributes xmlns="DR-GW"> <hook>{0,1}</hook> <mode>{0,1}</mode> <commtype>{0,1}</commtype> <priority>{0,1}</priority> <encryption>{0,1}</encryption> <ambienceListen>{0,1}</ambienceListen> <req2speak>{0,1}</req2speak> <demandPriority>{0,1}</demandPriority> </attributes></pre>				
Source	<code><xs:element name="attributes" type="typeCallAttributes" minOccurs="0" /></code>				

Element typeCallAttributes / hook

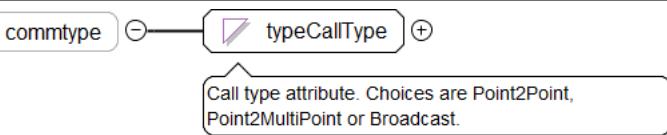
Namespace	DR-GW				
Diagram	<p>The diagram shows an attribute named <code>hook</code> with the type <code>xs:boolean</code>. A callout box below the type states: "Built-in primitive type. It defines the boolean values true and false."</p>				
Type	<code>xs:boolean</code>				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><xs:element name="hook" type="xs:boolean" minOccurs="0" /></code>				

Element typeCallAttributes / mode

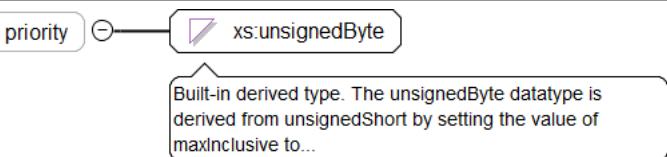
Namespace	DR-GW
Diagram	<p>The diagram shows an attribute named <code>mode</code> with the type <code>typeCallMode</code>. A callout box below the type states: "Call mode attribute. Choices are simplex or duplex."</p>

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

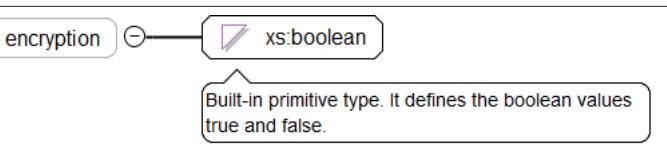
Element typeCallAttributes / commtype

Namespace	DR-GW
Diagram	 <p>The diagram shows a relationship between two elements: 'commtype' and 'typeCallType'. An association line connects them, with a multiplicity of 0..1 on the 'commtype' side and * on the 'typeCallType' side. A callout box points to the 'typeCallType' side of the association, containing the text: 'Call type attribute. Choices are Point2Point, Point2MultiPoint or Broadcast.'</p>
Type	typeCallType
Properties	content: simple minOccurs: 0
Facets	enumeration p2p enumeration p2mp enumeration bcast
Source	<code><xss:element name="commtype" type="typeCallType" minOccurs="0" /></code>

Element typeCallAttributes / priority

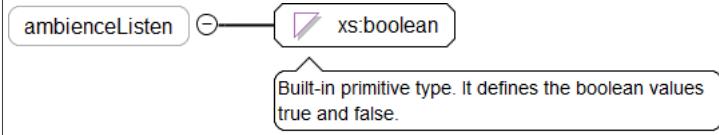
Namespace	DR-GW
Diagram	 <p>The diagram shows a relationship between 'priority' and 'xs:unsignedByte'. An association line connects them, with a multiplicity of 0..1 on the 'priority' side and * on the 'xs:unsignedByte' side. A callout box points to the 'xs:unsignedByte' side of the association, containing the text: 'Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...'</p>
Type	xs:unsignedByte
Properties	content: simple minOccurs: 0 default: 1
Source	<code><xss:element name="priority" type="xs:unsignedByte" default="1" minOccurs="0" /></code>

Element typeCallAttributes / encryption

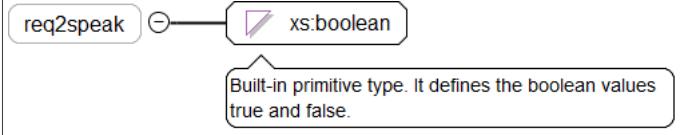
Namespace	DR-GW
Diagram	 <p>The diagram shows a relationship between 'encryption' and 'xs:boolean'. An association line connects them, with a multiplicity of 0..1 on the 'encryption' side and * on the 'xs:boolean' side. A callout box points to the 'xs:boolean' side of the association, containing the text: 'Built-in primitive type. It defines the boolean values true and false.'</p>
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: true
Source	<code><xss:element name="encryption" type="xs:boolean" default="true" minOccurs="0" /></code>

Element typeCallAttributes / ambienceListen

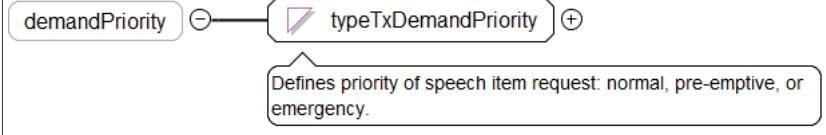
Namespace	DR-GW
-----------	-------

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

Element typeCallAttributes / req2speak

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>default: 1</p>
Source	<code><xs:element name="req2speak" type="xs:boolean" default="1" minOccurs="0" /></code>

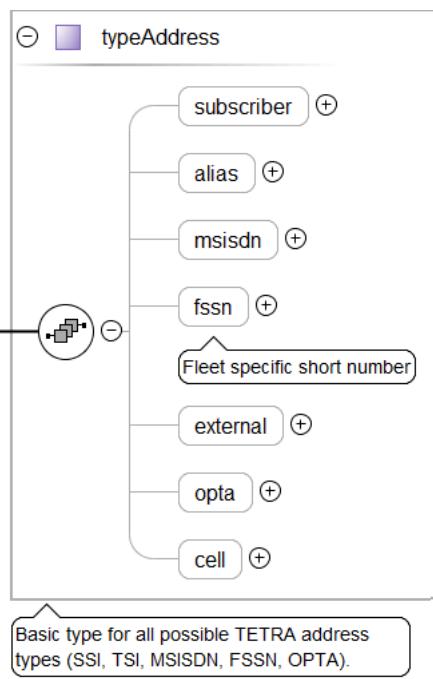
Element typeCallAttributes / demandPriority

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

Element typeCallEvent / callingParty

Namespace	DR-GW
-----------	-------

Diagram



Type	typeAddress
Properties	content: complex minOccurs: 0
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<pre> <callingParty xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </callingParty> </pre>
Source	<xss:element name="callingParty" type="typeAddress" minOccurs="0" />

Element typeCallEvent / calledParty

Namespace	DR-GW
-----------	-------

Diagram	<p>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</p>				
Type	typeAddress				
Properties	<table border="1"> <tr> <td>content:</td><td>complex</td></tr> <tr> <td>minOccurs:</td><td>0</td></tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}				
Children	alias, cell, external, fssn, msisdn, opta, subscriber				
Instance	<pre><calledParty xmlns="DR-GW"> <subscriber>{0..1}</subscriber> <alias>{0..1}</alias> <msisdn>{0..1}</msisdn> <fssn>{0..1}</fssn> <external>{0..1}</external> <opta>{0..1}</opta> <cell>{0..1}</cell> </calledParty></pre>				
Source	<code><xs:element name="calledParty" type="typeAddress" minOccurs="0" /></code>				

Element typeCallEvent / disconnectCause

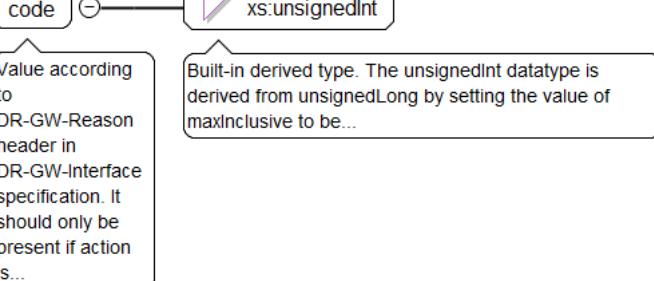
Namespace	DR-GW		
Diagram			
Type	typeDisconnectCause		
Properties	<table border="1"> <tr> <td>content:</td><td>complex</td></tr> </table>	content:	complex
content:	complex		

	minOccurs: 0
Model	protocol , code , text{0,1}
Children	code, protocol, text
Instance	<disconnectCause xmlns="DR-GW"> <protocol>{1,1}</protocol> <code>{1,1}</code> <text>{0,1}</text> </disconnectCause>
Source	<xss:element name="disconnectCause" type="typeDisconnectCause" minOccurs="0"/>

Element typeDisconnectCause / protocol

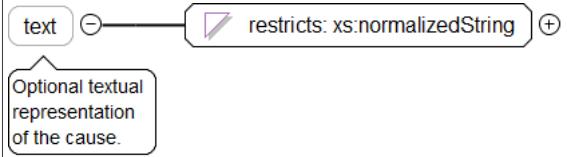
Namespace	DR-GW
Diagram	
Type	restriction of xs:normalizedString
Properties	content: simple
Facets	enumeration DR-GW enumeration TCS-API
Source	<xss:element name="protocol"> <xss:simpleType> <xss:restriction base="xs:normalizedString"> <xss:enumeration value="DR-GW"/> <xss:enumeration value="TCS-API"/> </xss:restriction> <br < xss:simpletype><br=""></br <> </xss:element>

Element typeDisconnectCause / code

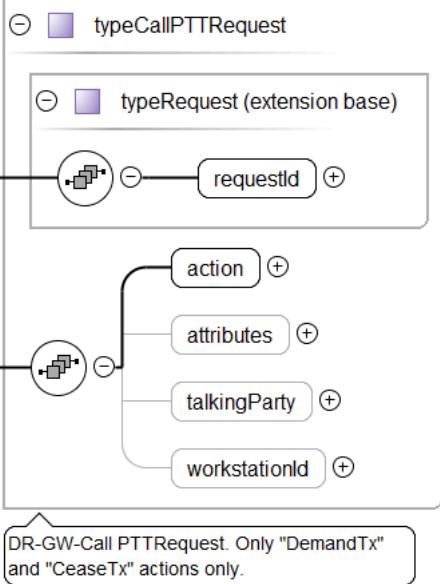
Namespace	DR-GW
Annotations	Value according to DR-GW-Reason header in DR-GW-Interface specification. It should only be present if action is "disconnected" and holds the reason for call disconnection.
Diagram	
Type	xs:unsignedInt
Properties	content: simple
Source	<xss:element name="code" type="xs:unsignedInt"> <xss:annotation> <xss: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.</xss:documentation> <br < xss:annotation><br=""></br <> </xss:element>

Element typeDisconnectCause / text

Namespace	DR-GW
Annotations	Optional textual representation of the cause.

Diagram	
Type	restriction of xs:normalizedString
Properties	content: simple minOccurs: 0
Facets	maxLength 80
Source	<pre><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></pre>

Element interfaceCall / pttRequest

Namespace	DR-GW
Diagram	 <p>DR-GW-Call PTTRequest. Only "DemandTx" and "CeaseTx" actions only.</p>
Type	typeCallPTTRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeCallPTTRequest
Properties	content: complex
Model	requestId , action , attributes{0,1} , talkingParty{0,1} , workstationId{0,1}
Children	action, attributes, requestId, talkingParty, workstationId
Instance	<pre><pttRequest xmlns="DR-GW"> <requestId>{1,1}</requestId> <action>{1,1}</action> <attributes>{0,1}</attributes> <talkingParty>{0,1}</talkingParty> <workstationId>{0,1}</workstationId> </pttRequest></pre>
Source	<pre><xs:element name="pttRequest" type="typeCallPTTRequest" /></pre>

Element typeCallPTTRequest / action

Namespace	DR-GW
-----------	-------

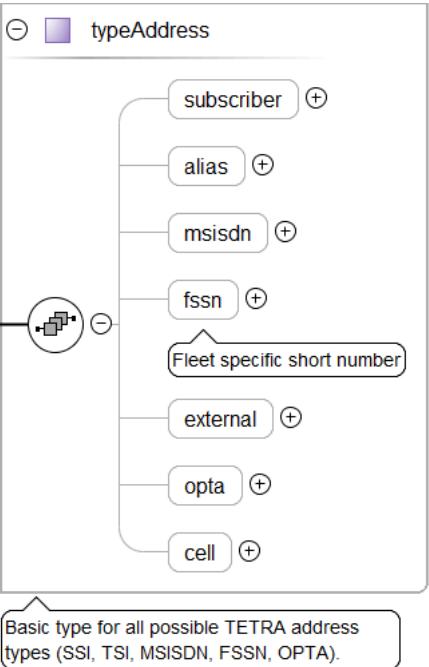
Diagram							
Type	typeActionPTTRequest						
Properties	content: simple						
Facets	<table> <tr> <td>enumeration</td> <td>demandtx</td> <td>This method can be used to request a speech item for a connected call.</td> </tr> <tr> <td>enumeration</td> <td>ceasetx</td> <td>This method is used to inform the system that the speech item is not needed any more.</td> </tr> </table>	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.
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	<code><xss:element name="action" type="typeActionPTTRequest"/></code>						

Element typeCallPTTRequest / attributes

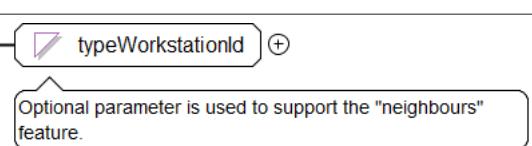
Namespace	DR-GW				
Diagram					
Type	typeCallAttributes				
Properties	<table> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	hook{0,1} , mode{0,1} , commtype{0,1} , priority{0,1} , encryption{0,1} , ambienceListen{0,1} , req2speak{0,1} , demandPriority{0,1}				
Children	ambienceListen, commtype, demandPriority, encryption, hook, mode, priority, req2speak				
Instance	<pre> <attributes xmlns="DR-GW"> <hook>{0,1}</hook> <mode>{0,1}</mode> <commtype>{0,1}</commtype> <priority>{0,1}</priority> <encryption>{0,1}</encryption> <ambienceListen>{0,1}</ambienceListen> <req2speak>{0,1}</req2speak> <demandPriority>{0,1}</demandPriority> </attributes> </pre>				
Source	<code><xss:element name="attributes" type="typeCallAttributes" minOccurs="0"/></code>				

Element typeCallPTTRequest / talkingParty

Namespace	DR-GW
-----------	-------

Diagram					
Type	typeAddress				
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> <tr> <td style="padding: 2px;">minOccurs:</td><td style="padding: 2px;">0</td></tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}				
Children	alias, cell, external, fssn, msisdn, opta, subscriber				
Instance	<pre><talkingParty xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </talkingParty></pre>				
Source	<code><xs:element name="talkingParty" type="typeAddress" minOccurs="0" /></code>				

Element typeCallPTTRequest / workstationId

Namespace	DR-GW				
Diagram					
Type	typeWorkstationId				
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">simple</td></tr> <tr> <td style="padding: 2px;">minOccurs:</td><td style="padding: 2px;">0</td></tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><xs:element name="workstationId" type="typeWorkstationId" minOccurs="0" /></code>				

Element interfaceCall / keyExchange

Namespace	DR-GW
-----------	-------

Diagram	<pre> classDiagram typeCallKeyExchange < -- typeRequest typeCallKeyExchange "0..1" --> "1..1" requestId typeCallKeyExchange "0..1" --> "1..1" action note over typeCallKeyExchange: For triggering the group key exchange. Key exchange events are sent in Call_KeyXEvent. </pre>
Type	typeCallKeyExchange
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeCallKeyExchange
Properties	content: complex
Model	requestId , action
Children	action, requestId
Instance	<pre> <keyExchange xmlns="DR-GW"> <requestId>{1,1}</requestId> <action>{1,1}</action> </keyExchange> </pre>
Source	<code><xss:element name="keyExchange" type="typeCallKeyExchange" /></code>

Element typeCallKeyExchange / action

Namespace	DR-GW				
Diagram	<pre> classDiagram typeKeyExchangeAction "0..1" --> "1..1" action note over typeKeyExchangeAction: Action type for key exchange request. </pre>				
Type	typeKeyExchangeAction				
Properties	content: simple				
Facets	<table border="1"> <tr> <td>enumeration</td> <td>start</td> </tr> <tr> <td>enumeration</td> <td>stop</td> </tr> </table>	enumeration	start	enumeration	stop
enumeration	start				
enumeration	stop				
Source	<code><xss:element name="action" type="typeKeyExchangeAction" /></code>				

Element interfaceCall / response

Namespace	DR-GW
Diagram	<pre> classDiagram typeResponse "0..1" --> "1..1" requestId typeResponse "0..1" --> "1..1" result note over typeResponse: Response contains result of execution of any method. </pre>
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result

Instance	<pre><response xmlns="DR-GW"> <requestId>{1,1}</requestId> <result>{1,1}</result> </response></pre>
Source	<pre><xs:element name="response" type="typeResponse" /></pre>

Element typeResponse / requestId

Namespace	DR-GW
Diagram	<p>A UML class diagram fragment showing the definition of the <code>requestId</code> element. It consists of a single class node labeled <code>requestId</code> with an outgoing association line ending in a hollow circle. This line is connected to a rectangle containing the text <code>xs:unsignedLong</code>. A callout box below the association line contains the text: "Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of..."</p>
Type	<code>xs:unsignedLong</code>
Properties	content: simple
Source	<pre><xs:element name="requestId" type="xs:unsignedLong" /></pre>

Element typeResponse / result

Namespace	DR-GW
Diagram	<p>A UML class diagram fragment showing the definition of the <code>result</code> element. It consists of a class node labeled <code>result</code> with an outgoing association line ending in a hollow circle. This line is connected to a class node labeled <code>typeResult</code>, which is enclosed in a rounded rectangle. Inside the <code>typeResult</code> box, there are three association lines originating from the <code>typeResult</code> node: one to a node labeled <code>responseCode</code> with multiplicity '+', one to a node labeled <code>sourceSystem</code> with multiplicity '+', and one to a node labeled <code>result</code> with multiplicity '+'. A callout box below the <code>typeResult</code> box contains the text: "Common result values used in every response and optional specific subsystem result codes."</p>
Type	<code>typeResult</code>
Properties	content: complex
Model	<code>responseCode , sourceSystem{0,1} , result{0,1}</code>
Children	<code>responseCode, result, sourceSystem</code>
Instance	<pre><result xmlns="DR-GW"> <responseCode>{1,1}</responseCode> <sourceSystem>{0,1}</sourceSystem> <result>{0,1}</result> </result></pre>
Source	<pre><xs:element name="result" type="typeResult" /></pre>

Element interfaceCall / selectEvent

Namespace	DR-GW
-----------	-------

Diagram	<pre> classDiagram class typeEvent { <<extension base>> requestId *--> selectEvent result *--> target } selectEvent --> typeEvent target --> typeEvent note over typeEvent: The event informs about the actual state of the selection requested before using the "select" request. </pre>
Type	typeCallSelectEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeCallSelectEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , target
Children	requestId, result, target
Instance	<pre> <selectEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <target>{1,1}</target> </selectEvent> </pre>
Source	<xss:element name="selectEvent" type="typeCallSelectEvent"/>

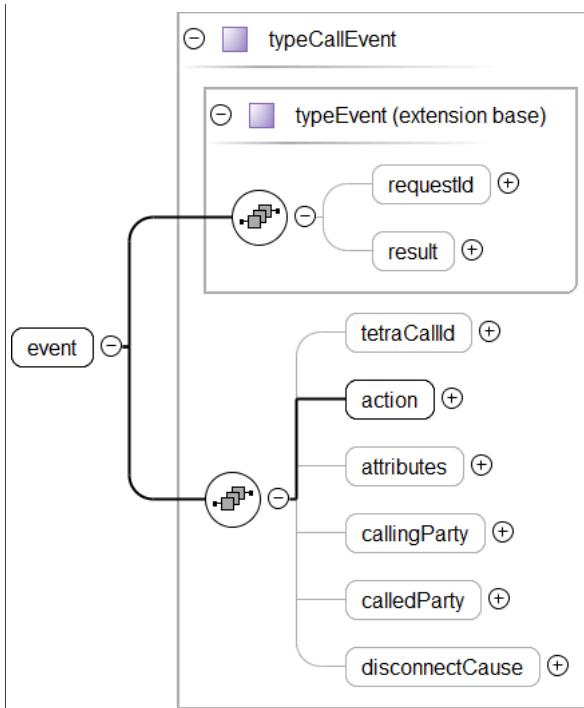
Element typeCallSelectEvent / target

Namespace	DR-GW
Diagram	<pre> classDiagram class typeSelection { level *--> target target *--> typeSelection } </pre>
Type	typeSelection
Properties	content: complex
Model	level , target
Children	level, target
Instance	<pre> <target xmlns="DR-GW"> <level>{1,1}</level> <target>{1,1}</target> </target> </pre>
Source	<xss:element name="target" type="typeSelection"/>

Element interfaceCall / event

Namespace	DR-GW
-----------	-------

Diagram



Type	typeCallEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent <ul style="list-style-type: none"> • typeCallEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , tetraCallId{0,1} , action , attributes{0,1} , callingParty{0,1} , calledParty{0,1} , disconnectCause{0,1}
Children	action, attributes, calledParty, callingParty, disconnectCause, requestId, result, tetraCallId
Instance	<pre> <event xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</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> </event> </pre>
Source	<code><xss:element name="event" type="typeCallEvent" /></code>

Element interfaceCall / pttEvent

Namespace	DR-GW
-----------	-------

Diagram	
Type	typeCallPTTEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeCallPTTEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , tetraCallId{0,1} , (granted ceased wait)
Children	ceased, granted, requestId, result, tetraCallId, wait
Instance	<pre><pttEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <tetraCallId>{0,1}</tetraCallId> <granted>{1,1}</granted> <ceased>{1,1}</ceased> <wait>{1,1}</wait> </pttEvent></pre>
Source	<code><xs:element name="pttEvent" type="typeCallPTTEvent" /></code>

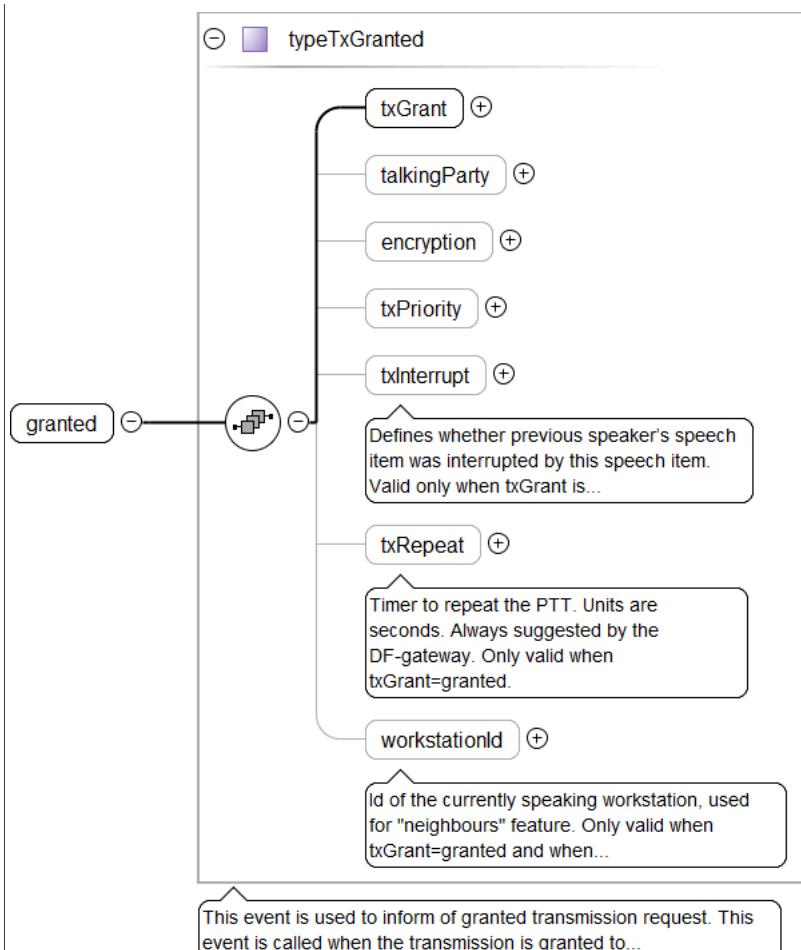
Element typeCallPTTEvent / tetraCallId

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

Element typeCallPTTEvent / granted

Namespace	DR-GW
-----------	-------

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	<pre> <granted xmlns="DR-GW"> <txGrant>{1,1}</txGrant> <talkingParty>{0,1}</talkingParty> <encryption>{0,1}</encryption> <txPriority>{0,1}</txPriority> <txInterrupt>{0,1}</txInterrupt> <txRepeat>{0,1}</txRepeat> <workstationId>{0,1}</workstationId> </granted></pre>
Source	<xss:element name="granted" type="typeTxGranted" />

Element typeTxGranted / txGrant

Namespace	DR-GW								
Diagram									
Type	typeTxGrant								
Properties	content: simple								
Facets	<table border="1"> <tbody> <tr> <td>enumeration</td><td>granted</td></tr> <tr> <td>enumeration</td><td>notGranted</td></tr> <tr> <td>enumeration</td><td>queued</td></tr> <tr> <td>enumeration</td><td>granted2another</td></tr> </tbody> </table>	enumeration	granted	enumeration	notGranted	enumeration	queued	enumeration	granted2another
enumeration	granted								
enumeration	notGranted								
enumeration	queued								
enumeration	granted2another								

Source	<code><xss:element name="txGrant" type="typeTxGrant" /></code>
--------	--

Element typeTxGranted / talkingParty

Namespace	DR-GW				
Diagram	<pre> classDiagram typeAddress { subscriber alias msisdn fssn external opta cell } talkingParty { <<dr-gw>> <subscriber> <alias> <msisdn> <fssn> <external> <opta> <cell> } typeAddress "0..1" o-- "0..1" talkingParty </pre> <p>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</p>				
Type	typeAddress				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}				
Children	alias, cell, external, fssn, msisdn, opta, subscriber				
Instance	<pre> <talkingParty xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </talkingParty> </pre>				
Source	<code><xss:element name="talkingParty" type="typeAddress" minOccurs="0" /></code>				

Element typeTxGranted / encryption

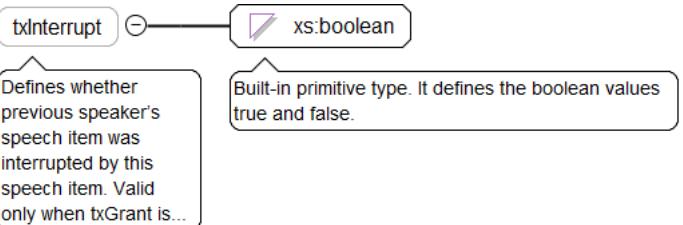
Namespace	DR-GW						
Diagram	<pre> classDiagram xs:boolean encryption { <<dr-gw>> <xs:boolean> } encryption "0..1" o-- "0..1" xs:boolean </pre> <p>Built-in primitive type. It defines the boolean values true and false.</p>						
Type	xs:boolean						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>default:</td> <td>true</td> </tr> </table>	content:	simple	minOccurs:	0	default:	true
content:	simple						
minOccurs:	0						
default:	true						
Source	<code><xss:element name="encryption" type="xs:boolean" default="true" minOccurs="0" /></code>						

Element typeTxGranted / txPriority

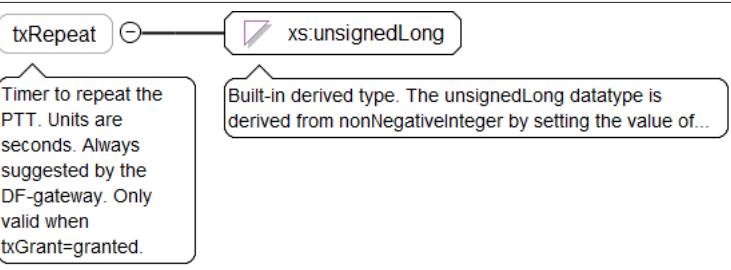
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeTxPriority
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>default: normal</p>
Facets	<p>enumeration normal</p> <p>enumeration emergency</p>
Source	<code><xss:element name="txPriority" type="typeTxPriority" minOccurs="0" default="normal" /></code>

Element typeTxGranted / txInterrupt

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

Element typeTxGranted / txRepeat

Namespace	DR-GW
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	<p>content: simple</p> <p>minOccurs: 0</p> <p>default: 0</p>
Source	<code><xss:element name="txRepeat" type="xs:unsignedLong" minOccurs="0" default="0"> <xss:annotation></code>

```

<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				
Annotations	<p>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.</p>				
Diagram	<p>Diagram illustrating the relationship between the element and its type:</p> <ul style="list-style-type: none"> The element <code>workstationId</code> is shown as a rounded rectangle. The type <code>xs:normalizedString</code> is shown as a rectangle with a purple border. A line connects the two, indicating that <code>workstationId</code> is of type <code>xs:normalizedString</code>. <p>Annotations for the diagram:</p> <ul style="list-style-type: none"> <code>workstationId</code>: Id of the currently speaking workstation, used for "neighbours" feature. Only valid when txGrant=granted and when... <code>xs:normalizedString</code>: Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of... 				
Type	<code>xs:normalizedString</code>				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<pre> <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> </pre>				

Element typeCallPTTEvent / ceased

Namespace	DR-GW		
Annotations	<p>This event is used to inform that transmission is ceased and nobody has the speech item.</p>		
Diagram	<p>Diagram illustrating the relationship between the element and its type:</p> <ul style="list-style-type: none"> The element <code>ceased</code> is shown as a rounded rectangle. The type <code>typeEmpty</code> is shown as a rectangle with a purple border. A line connects the two, indicating that <code>ceased</code> is of type <code>typeEmpty</code>. <p>Annotations for the diagram:</p> <ul style="list-style-type: none"> <code>ceased</code>: This event is used to inform that transmission is ceased and nobody has the speech item. <code>typeEmpty</code>: Explicit type specification for elements that shall be empty. 		
Type	<code>typeEmpty</code>		
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> </table>	content:	complex
content:	complex		
Source	<pre> <xs:element name="ceased" type="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> </pre>		

Element typeCallPTTEvent / wait

Namespace	DR-GW
Annotations	<p>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.</p>

Diagram	
Type	typeEmpty
Properties	content: complex
Source	<pre><xss:element name="wait" type="typeEmpty"> <xss:annotation> <xss: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.</xss:documentation> </xss:annotation> </xss:element></pre>

Element interfaceCall / unitInEmergencyEvent

Namespace	DR-GW
Diagram	
Type	typeCallUnitInEmergencyEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeCallUnitInEmergencyEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , tetraCallId{0,1} , unitInEmg , unitInEmgType , emgInfo , tstamp
Children	emgInfo, group, requestId, result, tetraCallId, tstamp, unitInEmg, unitInEmgType
Instance	<pre><unitInEmergencyEvent xmlns="DR-GW"> <requestId>{0,1}</requestId></pre>

	<pre> <result>{0,1}</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> </unitInEmergencyEvent> </pre>
Source	<pre><xs:element name="unitInEmergencyEvent" type="typeCallUnitInEmergencyEvent" /></pre>

Element typeCallUnitInEmergencyEvent / group

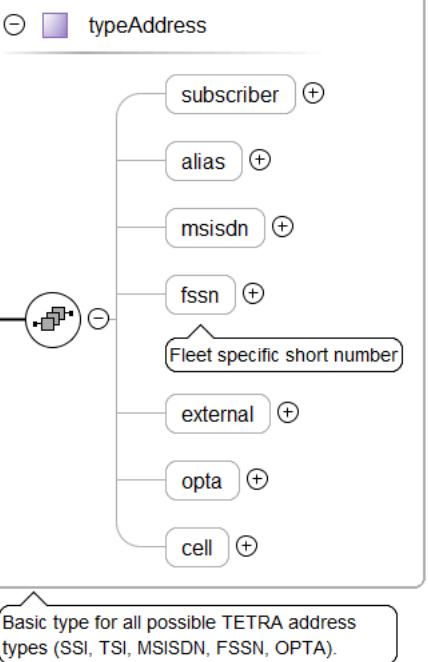
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre> <group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group> </pre>
Source	<pre><xs:element name="group" type="typeSubscriberAddress" /></pre>

Element typeCallUnitInEmergencyEvent / tetraCallId

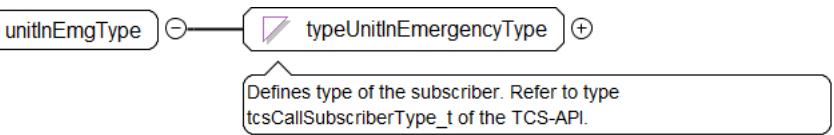
Namespace	DR-GW				
Diagram					
Type	xs:unsignedLong				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<pre><xs:element name="tetraCallId" type="xs:unsignedLong" minOccurs="0" /></pre>				

Element typeCallUnitInEmergencyEvent / unitInEmg

Namespace	DR-GW
-----------	-------

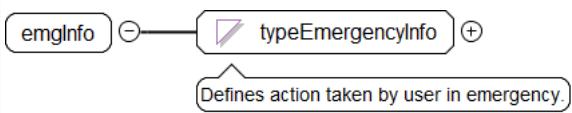
Diagram	 <p>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</p>
Type	typeAddress
Properties	content: complex
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<pre><unitInEmg xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </unitInEmg></pre>
Source	<code><xss:element name="unitInEmg" type="typeAddress" /></code>

Element typeCallUnitInEmergencyEvent / unitInEmgType

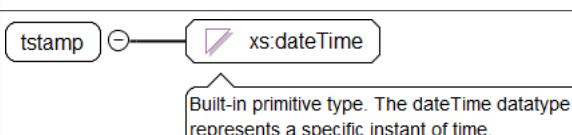
Namespace	DR-GW
Diagram	 <p>Defines type of the subscriber. Refer to type tcsCallSubscriberType_t of the TCS-API.</p>
Type	typeUnitInEmergencyType
Properties	content: simple
Facets	enumeration dummy enumeration ms enumeration g4wif enumeration external enumeration ws
Source	<code><xss:element name="unitInEmgType" type="typeUnitInEmergencyType" /></code>

Element typeCallUnitInEmergencyEvent / emgInfo

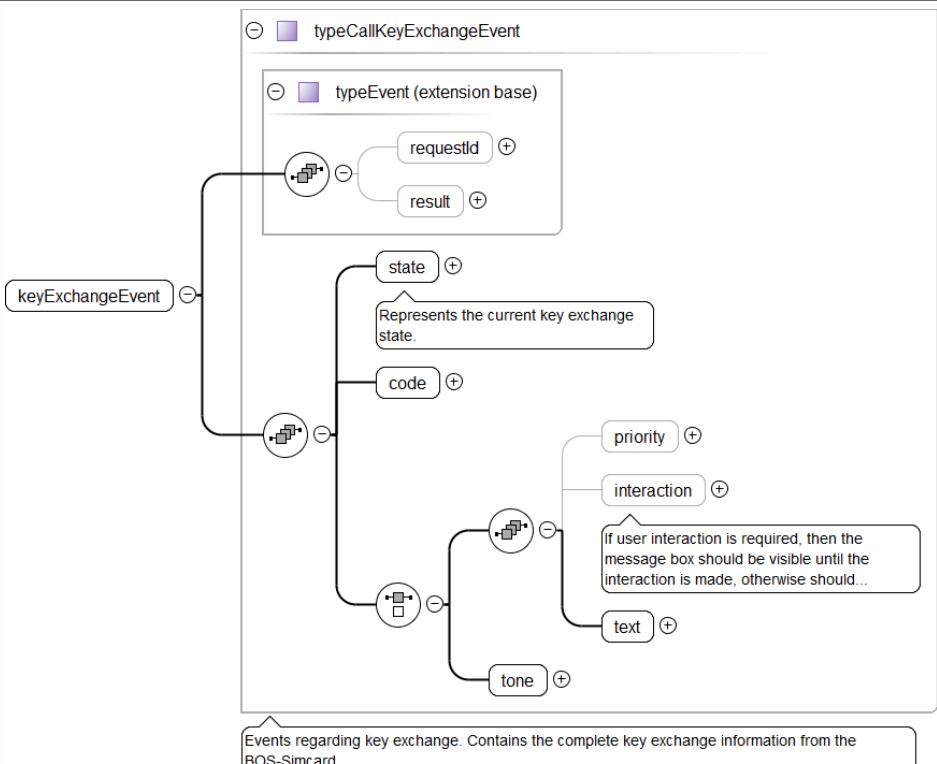
Namespace	DR-GW
-----------	-------

Diagram	 Defines action taken by user in emergency.
Type	typeEmergencyInfo
Properties	content: simple
Facets	enumeration addTx enumeration add enumeration ceased enumeration demandTx enumeration removed enumeration emergencyCallDisconnected
Source	<xs:element name="emgInfo" type="typeEmergencyInfo"/>

Element typeCallUnitInEmergencyEvent / tstamp

Namespace	DR-GW
Diagram	 Built-in primitive type. The dateTime datatype represents a specific instant of time.
Type	xs:dateTime
Properties	content: simple
Source	<xs:element name="tstamp" type="xs:dateTime"/>

Element interfaceCall / keyExchangeEvent

Namespace	DR-GW
Diagram	 Events regarding key exchange. Contains the complete key exchange information from the BOS-Simcard.
Type	typeCallKeyExchangeEvent
Type hierarchy	• typeEvent

	<ul style="list-style-type: none"> • typeCallKeyExchangeEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , state , code , ((priority{0,1} , interaction{0,1} , text) tone)
Children	code, interaction, priority, requestId, result, state, text, tone
Instance	<pre><keyExchangeEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</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> </keyExchangeEvent></pre>
Source	<code><xss:element name="keyExchangeEvent" type="typeCallKeyExchangeEvent" /></code>

Element typeCallKeyExchangeEvent / state

Namespace	DR-GW									
Annotations	Represents the current key exchange state.									
Diagram	<pre> classDiagram state "–" typeKeyExchangeState state --> stateNote typeKeyExchangeState --> stateNote stateNote : Represents the current key exchange state. </pre> <p>The diagram shows a class named 'state' connected to another class named 'typeKeyExchangeState' via a directed association. A callout box labeled 'Represents the current key exchange state.' points to the 'state' class.</p>									
Type	typeKeyExchangeState									
Properties	content: simple									
Facets	<table> <tr> <td>enumeration</td> <td>keyValid</td> <td>current key is valid, no user action required.</td> </tr> <tr> <td>enumeration</td> <td>keyInvalid</td> <td>Key invalid, user must request key exchange.</td> </tr> <tr> <td>enumeration</td> <td>keyExchangeInProgress</td> <td>Key exchange in progress, user may abort exchange or wait until it gets finished.</td> </tr> </table>	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.
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	<pre><xss:element name="state" type="typeKeyExchangeState"> <xss:annotation> <xss:documentation>Represents the current key exchange state.</xss:documentation> </xss:annotation> </xss:element></pre>									

Element typeCallKeyExchangeEvent / code

Namespace	DR-GW
Diagram	<pre> classDiagram code "–" typeKeyExchangeCode code --> codeNote typeKeyExchangeCode --> codeNote codeNote : See "Table 5.3: Status words of the commands" of the E-to-E Encryption SIM-ME Interface (Version 4.0.5) for all... </pre> <p>The diagram shows a class named 'code' connected to another class named 'typeKeyExchangeCode' via a directed association. A callout box labeled 'See "Table 5.3: Status words of the commands" of the E-to-E Encryption SIM-ME Interface (Version 4.0.5) for all...' points to the 'code' class.</p>
Type	typeKeyExchangeCode
Properties	content: simple
Facets	length 2
Source	<code><xss:element name="code" type="typeKeyExchangeCode" /></code>

Element typeCallKeyExchangeEvent / priority

Namespace	DR-GW
Diagram	<pre> classDiagram priority "–" typeKeyExchangeTextPriority priority --> priorityNote typeKeyExchangeTextPriority --> priorityNote priorityNote : Defines the priority of the KeyExchange information. </pre> <p>The diagram shows a class named 'priority' connected to another class named 'typeKeyExchangeTextPriority' via a directed association. A callout box labeled 'Defines the priority of the KeyExchange information.' points to the 'priority' class.</p>
Type	typeKeyExchangeTextPriority

Properties	content: simple minOccurs: 0 default: normal
Facets	enumeration normal enumeration high
Source	<code><xs:element name="priority" type="typeKeyExchangeTextPriority" minOccurs="0" default="normal" /></code>

Element typeCallKeyExchangeEvent / interaction

Namespace	DR-GW
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	<code><xs:element name="interaction" type="xs:boolean" minOccurs="0" default="false"></code> <code> <xs:annotation></code> <code> <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></code> <code> </xs:annotation></code> <code></xs:element></code>

Element typeCallKeyExchangeEvent / text

Namespace	DR-GW
Diagram	
Type	typeKeyExchangeText
Properties	content: simple
Facets	maxLength 100
Source	<code><xs:element name="text" type="typeKeyExchangeText" /></code>

Element typeCallKeyExchangeEvent / tone

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

Element drgw / session

Namespace	DR-GW
Diagram	<pre> classDiagram class session { <<Type interfaceSession>> } class interfaceSession { login logout supervise check response loginEvent superviseEvent } session < -- interfaceSession </pre> <p>DR-GW-Session. In order to use the rest of DR-GW interface the DF-Client must establish a DR-GW session and maintain it...</p>
Type	interfaceSession
Properties	content: complex
Model	login logout supervise check response loginEvent superviseEvent
Children	check, login, loginEvent, logout, response, supervise, superviseEvent
Instance	<pre> <session xmlns="DR-GW"> <login>{1,1}</login> <logout>{1,1}</logout> <supervise>{1,1}</supervise> <check>{1,1}</check> <response>{1,1}</response> <loginEvent>{1,1}</loginEvent> <superviseEvent>{1,1}</superviseEvent> </session> </pre>
Source	<xs:element name="session" type="interfaceSession" />

Element interfaceSession / login

Namespace	DR-GW
Diagram	<pre> classDiagram class login class typeSessionLogin { <<typeRequest (extension base)>> } class typeRequest login < -- typeSessionLogin </pre> <p>Login procedure. The username, password and the complete authentication is done using mechanisms of the transport...</p>

Type	typeSessionLogin
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeSessionLogin
Properties	content: complex
Model	requestId , clientid , supervise{0,1} , version{0,1}
Children	clientid, requestId, supervise, version
Instance	<pre><login xmlns="DR-GW"> <requestId>{1,1}</requestId> <clientid>{1,1}</clientid> <supervise>{0,1}</supervise> <version>{0,1}</version> </login></pre>
Source	<code><xss:element name="login" type="typeSessionLogin"/></code>

Element typeSessionLogin / clientid

Namespace	DR-GW
Diagram	<p>A UML class diagram fragment showing a dependency relationship between two elements. On the left, there is a rounded rectangle labeled "clientid". A line with a hollow circle at the "clientid" end and a solid circle at the other end connects it to another element. This second element is a rounded rectangle containing a purple pencil icon and the text "xs:string". A callout bubble below this second element contains the text: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xs:string
Properties	content: simple
Source	<code><xss:element name="clientid" type="xs:string"/></code>

Element typeSessionLogin / supervise

Namespace	DR-GW						
Diagram	<p>A UML class diagram fragment showing a dependency relationship between two elements. On the left, there is a rounded rectangle labeled "supervise". A line with a hollow circle at the "supervise" end and a solid circle at the other end connects it to another element. This second element is a rounded rectangle containing a purple pencil icon and the text "typeSuperviseTimeout". A plus sign (+) is at the end of the line. A callout bubble below this second element contains the text: "Accepted supervise timeout values."</p>						
Type	typeSuperviseTimeout						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>default:</td> <td>60</td> </tr> </table>	content:	simple	minOccurs:	0	default:	60
content:	simple						
minOccurs:	0						
default:	60						
Facets	<table border="1"> <tr> <td>enumeration</td> <td>20</td> </tr> <tr> <td>enumeration</td> <td>30</td> </tr> <tr> <td>enumeration</td> <td>60</td> </tr> </table>	enumeration	20	enumeration	30	enumeration	60
enumeration	20						
enumeration	30						
enumeration	60						
Source	<code><xss:element name="supervise" type="typeSuperviseTimeout" default="60" minOccurs="0"/></code>						

Element typeSessionLogin / version

Namespace	DR-GW				
Diagram	<p>A UML class diagram fragment showing a dependency relationship between two elements. On the left, there is a rounded rectangle labeled "version". A line with a hollow circle at the "version" end and a solid circle at the other end connects it to another element. This second element is a rounded rectangle containing a purple pencil icon and the text "xs:string". A callout bubble below this second element contains the text: "Built-in primitive type. The string datatype represents character strings in XML."</p>				
Type	xs:string				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><xss:element name="version" type="xs:string" minOccurs="0"/></code>				

Element interfaceSession / logout

Namespace	DR-GW
Diagram	
Type	typeSessionLogout
Type hierarchy	<ul style="list-style-type: none"> • typeRequest <ul style="list-style-type: none"> • typeSessionLogout
Properties	content: complex
Model	requestId
Children	requestId
Instance	<pre><logout xmlns="DR-GW"> <requestId>{1,1}</requestId> </logout></pre>
Source	<pre><xss:element name="logout" type="typeSessionLogout" /></pre>

Element interfaceSession / supervise

Namespace	DR-GW
Diagram	
Type	typeSessionSupervise
Type hierarchy	<ul style="list-style-type: none"> • typeRequest <ul style="list-style-type: none"> • typeSessionSupervise
Properties	content: complex
Model	requestId
Children	requestId
Instance	<pre><supervise xmlns="DR-GW"> <requestId>{1,1}</requestId> </supervise></pre>
Source	<pre><xss:element name="supervise" type="typeSessionSupervise" /></pre>

Element interfaceSession / check

Namespace	DR-GW
-----------	-------

Diagram	<pre> classDiagram typeSessionCheck < -- typeRequest typeSessionCheck "1..1" --> "1..1" check check "1..1" --> "1..1" requestId check "1..1" --> "1..1" clientid </pre> <p>To enable the DF-Client to check connectivity to DR-GW the client may use session check. The check requires the http...</p>
Type	typeSessionCheck
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeSessionCheck
Properties	content: complex
Model	requestId , clientid
Children	clientid, requestId
Instance	<pre> <check xmlns="DR-GW"> <requestId>{1,1}</requestId> <clientid>{1,1}</clientid> </check> </pre>
Source	<xs:element name="check" type="typeSessionCheck" />

Element typeSessionCheck / clientid

Namespace	DR-GW
Diagram	<pre> classDiagram clientid : xs:string </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Source	<xs:element name="clientid" type="xs:string" />

Element interfaceSession / response

Namespace	DR-GW
Diagram	<pre> classDiagram typeResponse "1..1" --> "1..1" response response "1..1" --> "1..1" requestId response "1..1" --> "1..1" result </pre> <p>Response contains result of execution of any method.</p>
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<pre> <response xmlns="DR-GW"> <requestId>{1,1}</requestId> <result>{1,1}</result> </response> </pre>

	</response>
Source	<xs:element name="response" type="typeResponse" />

Element interfaceSession / loginEvent

Namespace	DR-GW
Diagram	<pre> classDiagram typeSessionLoginEvent < -- typeEvent typeSessionLoginEvent { requestId result issi } loginEvent < -- typeSessionLoginEvent </pre>
Type	typeSessionLoginEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent <ul style="list-style-type: none"> • typeSessionLoginEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , issi{0,1}
Children	issi, requestId, result
Instance	<loginEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <issi>{0,1}</issi> </loginEvent>
Source	<xs:element name="loginEvent" type="typeSessionLoginEvent" />

Element typeSessionLoginEvent / issi

Namespace	DR-GW				
Diagram	<pre> classDiagram issi < -- xs:string </pre> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>				
Type	xs:string				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<xs:element name="issi" type="xs:string" minOccurs="0" />				

Element interfaceSession / superviseEvent

Namespace	DR-GW
Diagram	<pre> classDiagram typeSessionSuperviseEvent < -- typeEvent typeSessionSuperviseEvent { requestId result "superviseEvent" } superviseEvent < -- typeSessionSuperviseEvent </pre>
Type	typeSessionSuperviseEvent

Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeSessionSuperviseEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1}
Children	requestId, result
Instance	<pre><superviseEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> </superviseEvent></pre>
Source	<pre><xss:element name="superviseEvent" type="typeSessionSuperviseEvent" /></pre>

Element drgw / sds

Namespace	DR-GW
Diagram	<p>The diagram illustrates the <code>interfaceSds</code> element. It consists of a class named <code>sds</code> which has two associations: one to <code>Type</code> and one to <code>interfaceSds</code>. The <code>interfaceSds</code> class contains six operations: <code>send</code>, <code>sendReport</code>, <code>response</code>, <code>sendEvent</code>, <code>receiveEvent</code>, and <code>reportEvent</code>. A callout box below the diagram states: "DR-GW-Sds element. Use to send/receive SDS messages. Use only via SOAP."</p>
Type	interfaceSds
Properties	content: complex
Model	send sendReport response sendEvent receiveEvent reportEvent
Children	receiveEvent, reportEvent, response, send, sendEvent, sendReport
Instance	<pre><sds xmlns="DR-GW"> <send>{1,1}</send> <sendReport>{1,1}</sendReport> <response>{1,1}</response> <sendEvent>{1,1}</sendEvent> <receiveEvent>{1,1}</receiveEvent> <reportEvent>{1,1}</reportEvent> </sds></pre>
Source	<pre><xss:element name="sds" type="interfaceSds" /></pre>

Element interfaceSds / send

Namespace	DR-GW
Diagram	<p>The diagram illustrates the <code>typeSdsSend</code> element. It has a <code>send</code> association and two <code>sds</code> associations. One <code>sds</code> association points to a class named <code>typeRequest (extension base)</code>, which contains an attribute <code>requestId</code>. The other <code>sds</code> association points to another <code>sds</code> class.</p>

Type	typeSdsSend
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeSdsSend
Properties	content: complex
Model	requestId , sds
Children	requestId, sds
Instance	<pre><send xmlns="DR-GW"> <requestId>{1,1}</requestId> <sds>{1,1}</sds> </send></pre>
Source	<code><xss:element name="send" type="typeSdsSend" /></code>

Element typeSdsSend / sds

Namespace	DR-GW
Diagram	<pre> classDiagram class typeSds { protocol {0..1} sdsType {0..1} msgRef {0..1} report {0..1} sdsdata {0..1} source {0..1} target {0..1} forward {0..1} validity {0..1} tstamp {0..1} encryption {0..1} e2eegroup {0..1} } typeSds "0..1" --> "0..1" sds : </pre>
Type	typeSds
Properties	content: complex
Model	protocol{0,1} , sdsType , msgRef{0,1} , report{0,1} , sdsdata , source{0,1} , target , forward{0,1} , validity{0,1} , tstamp{0,1} , encryption{0,1} , e2eegroup{0,1}
Children	e2eegroup, encryption, forward, msgRef, protocol, report, sdsType, sdsdata, source, target, tstamp, validity
Instance	<pre><sds xmlns="DR-GW"> <protocol>{0,1}</protocol> <sdsType>{1,1}</sdsType> <msgRef>{0,1}</msgRef> <report>{0,1}</report> <sdsdata>{1,1}</sdsdata> <source>{0,1}</source> <target>{1,1}</target> <forward>{0,1}</forward> <validity>{0,1}</validity> <tstamp>{0,1}</tstamp> <encryption>{0,1}</encryption></pre>

	<e2eegroup>{0,1}</e2eegroup> </sds>
Source	<xs:element name="sds" type="typeSds" />

Element typeSds / protocol

Namespace	DR-GW				
Diagram	<pre> classDiagram class protocol { xs:unsignedByte } </pre> <p>Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...</p>				
Type	xs:unsignedByte				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<xs:element name="protocol" type="xs:unsignedByte" minOccurs="0" />				

Element typeSds / sdsType

Namespace	DR-GW																		
Diagram	<pre> classDiagram class sdsType { typeSdsType } </pre>																		
Type	typeSdsType																		
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple																
content:	simple																		
Facets	<table> <tr> <td>enumeration</td> <td>0</td> <td>SDS1.</td> </tr> <tr> <td>enumeration</td> <td>1</td> <td>SDS2.</td> </tr> <tr> <td>enumeration</td> <td>2</td> <td>SDS3.</td> </tr> <tr> <td>enumeration</td> <td>3</td> <td>SDS4.</td> </tr> <tr> <td>enumeration</td> <td>4</td> <td>SDS-TL.</td> </tr> <tr> <td>enumeration</td> <td>5</td> <td>Status.</td> </tr> </table>	enumeration	0	SDS1.	enumeration	1	SDS2.	enumeration	2	SDS3.	enumeration	3	SDS4.	enumeration	4	SDS-TL.	enumeration	5	Status.
enumeration	0	SDS1.																	
enumeration	1	SDS2.																	
enumeration	2	SDS3.																	
enumeration	3	SDS4.																	
enumeration	4	SDS-TL.																	
enumeration	5	Status.																	
Source	<xs:element name="sdsType" type="typeSdsType" />																		

Element typeSds / msgRef

Namespace	DR-GW				
Diagram	<pre> classDiagram class msgRef { xs:unsignedByte } </pre> <p>Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...</p>				
Type	xs:unsignedByte				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<xs:element name="msgRef" type="xs:unsignedByte" minOccurs="0" />				

Element typeSds / report

Namespace	DR-GW						
Diagram	<pre> classDiagram class report { typeReport } </pre>						
Type	typeReport						
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>default:</td> <td>none</td> </tr> </table>	content:	simple	minOccurs:	0	default:	none
content:	simple						
minOccurs:	0						
default:	none						

Facets	enumeration none enumeration delivery enumeration consume enumeration both
Source	<xs:element name="report" type="typeReport" default="none" minOccurs="0" />

Element typeSds / sdsdata

Namespace	DR-GW
Diagram	<p>2 ways of encoding the SDS. When sent from DF-Client to DF-Gateway at least one node must be present, otherwise it will...</p>
Type	typeSdsData
Properties	content: complex
Model	data{0,1} , hexdata{0,1} , hexdatalength{0,1}
Children	data, hexdata, hexdatalength
Instance	<sdsdata xmlns="DR-GW"> <data>{0,1}</data> <hexdata>{0,1}</hexdata> <hexdatalength>{0,1}</hexdatalength> </sdsdata>
Source	<xs:element name="sdsdata" type="typeSdsData" />

Element typeSdsData / data

Namespace	DR-GW
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple minOccurs: 0
Source	<xs:element name="data" type="xs:string" minOccurs="0" />

Element typeSdsData / hexdata

Namespace	DR-GW
Diagram	<p>Built-in primitive type. The hexBinary datatype represents arbitrary hex-encoded binary data.</p>
Type	xs:hexBinary
Properties	content: simple minOccurs: 0

Source	<code><xss:element name="hexdata" type="xs:hexBinary" minOccurs="0" /></code>
--------	---

Element typeSdsData / hexdatalength

Namespace	DR-GW				
Diagram	<p>The diagram shows a class named "hexdatalength" with a directed association line pointing to a class named "xs:integer". A callout box indicates that "xs:integer" is a built-in derived type derived from decimal by fixing the value of fractionDigits to 0.</p>				
Type	xs:integer				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><xss:element name="hexdatalength" type="xs:integer" minOccurs="0" /></code>				

Element typeSds / source

Namespace	DR-GW				
Diagram	<p>The diagram shows a class named "source" with a directed association line pointing to a class named "typeAddress". The "typeAddress" class contains several subclasses: "subscriber", "alias", "msisdn", "fssn", "external", "opta", and "cell". A callout box indicates that "typeAddress" is a basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</p>				
Type	typeAddress				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}				
Children	alias, cell, external, fssn, msisdn, opta, subscriber				
Instance	<pre><source xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </source></pre>				
Source	<code><xss:element name="source" type="typeAddress" minOccurs="0" /></code>				

Element typeSds / target

Namespace	DR-GW
-----------	-------

Diagram	<pre> classDiagram class typeAddress { subscriber alias msisdn fssn external opta cell } target --> typeAddress note over fssn: Fleet specific short number note over typeAddress: Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA) </pre>
Type	typeAddress
Properties	content: complex
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<pre> <target xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </target> </pre>
Source	<pre> <xss:element name="target" type="typeAddress" /> </pre>

Element typeSds / forward

Namespace	DR-GW
-----------	-------

Diagram	<p>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</p>				
Type	typeAddress				
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> <tr> <td style="padding: 2px;">minOccurs:</td><td style="padding: 2px;">0</td></tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}				
Children	alias, cell, external, fssn, msisdn, opta, subscriber				
Instance	<pre><forward xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </forward></pre>				
Source	<code><xss:element name="forward" type="typeAddress" minOccurs="0" /></code>				

Element typeSds / validity

Namespace	DR-GW				
Diagram	<p>Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...</p>				
Type	xs:unsignedByte				
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">simple</td></tr> <tr> <td style="padding: 2px;">minOccurs:</td><td style="padding: 2px;">0</td></tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><xss:element name="validity" type="xs:unsignedByte" minOccurs="0" /></code>				

Element typeSds / tstamp

Namespace	DR-GW
Diagram	<p>Built-in primitive type. The dateTime datatype represents a specific instant of time.</p>

Type	xs:dateTime
Properties	content: simple minOccurs: 0
Source	<xs:element name="tstamp" type="xs:dateTime" minOccurs="0"/>

Element typeSds / encryption

Namespace	DR-GW
Diagram	<pre> graph LR encryption[encryption] --> xsboolean[xs:boolean] </pre> <p>Built-in primitive type. It defines the boolean values true and false.</p>
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: true
Source	<xs:element name="encryption" type="xs:boolean" default="true" minOccurs="0"/>

Element typeSds / e2eegroup

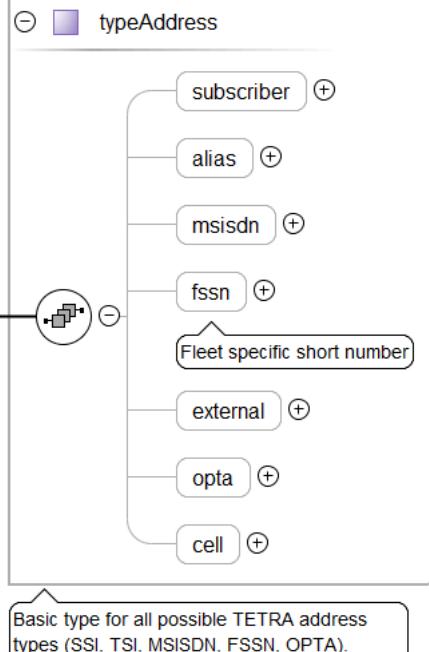
Namespace	DR-GW
Diagram	<pre> graph LR e2eegroup[e2eegroup] --> typeSubscriberAddress[typeSubscriberAddress] typeSubscriberAddress --> ssi[ssi] typeSubscriberAddress --> tsi[tsi] </pre>
Type	typeSubscriberAddress
Properties	content: complex minOccurs: 0
Model	ssi tsi
Children	ssi, tsi
Instance	<e2eegroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </e2eegroup>
Source	<xs:element name="e2eegroup" type="typeSubscriberAddress" minOccurs="0"/>

Element interfaceSds / sendReport

Namespace	DR-GW
Diagram	<pre> graph LR sendReport[sendReport] --> typeSdsSendReport[typeSdsSendReport] typeSdsSendReport --> typeRequest[typeRequest extension base] typeRequest --> requestId[requestId] typeRequest --> target[target] typeRequest --> msgRef[msgRef] typeRequest --> deliveryStatus[deliveryStatus] </pre>

Type	typeSdsSendReport
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeSdsSendReport
Properties	content: complex
Model	requestId , target , msgRef , deliveryStatus
Children	deliveryStatus, msgRef, requestId, target
Instance	<pre><sendReport xmlns="DR-GW"> <requestId>{1,1}</requestId> <target>{1,1}</target> <msgRef>{1,1}</msgRef> <deliveryStatus>{1,1}</deliveryStatus> </sendReport></pre>
Source	<code><xs:element name="sendReport" type="typeSdsSendReport"/></code>

Element typeSdsSendReport / target

Namespace	DR-GW
Diagram	 <p>The diagram illustrates the structure of the typeAddress element. It features a central typeAddress node with a minus sign (-) at the top left. Seven attributes are listed to its right, each preceded by a plus sign (+): subscriber, alias, msisdn, fssn, external, opta, and cell. A line connects the target node to the typeAddress node. A callout box at the bottom right specifies: "Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA)."</p>
Type	typeAddress
Properties	content: complex
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<pre><target xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </target></pre>
Source	<code><xs:element name="target" type="typeAddress"/></code>

Element typeSdsSendReport / msgRef

Namespace	DR-GW
-----------	-------

Diagram	
Type	xs:unsignedByte
Properties	content: simple
Source	<code><xs:element name="msgRef" type="xs:unsignedByte"/></code>

Element typeSdsSendReport / deliveryStatus

Namespace	DR-GW
Diagram	
Type	xs:unsignedByte
Properties	content: simple
Source	<code><xs:element name="deliveryStatus" type="xs:unsignedByte"/></code>

Element interfaceSds / response

Namespace	DR-GW
Diagram	
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<code><response xmlns="DR-GW"> <requestId>{1,1}</requestId> <result>{1,1}</result> </response></code>
Source	<code><xs:element name="response" type="typeResponse"/></code>

Element interfaceSds / sendEvent

Namespace	DR-GW
-----------	-------

Diagram	<pre> classDiagram typeEvent < -- typeSdsSendEvent typeSdsSendEvent "0..1" o--> sendEvent typeSdsSendEvent "0..1" o--> requestId typeSdsSendEvent "0..1" o--> result typeSdsSendEvent "0..1" o--> msgRef typeSdsSendEvent "0..1" o--> sds note over msgRef: An message reference is returned in the response for later message identification in case delivery and/or consume... </pre>
Type	typeSdsSendEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeSdsSendEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , msgRef{0,1} , sds
Children	msgRef, requestId, result, sds
Instance	<pre> <sendEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <msgRef>{0,1}</msgRef> <sds>{1,1}</sds> </sendEvent> </pre>
Source	<xss:element name="sendEvent" type="typeSdsSendEvent" />

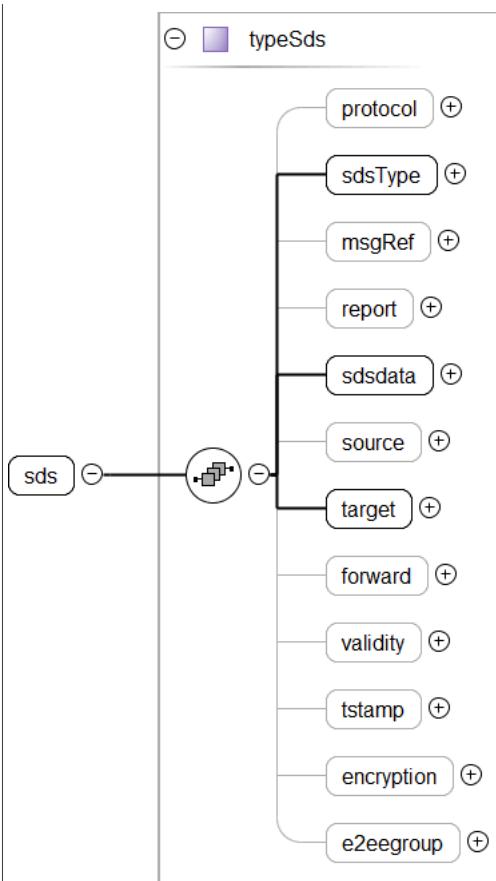
Element typeSdsSendEvent / msgRef

Namespace	DR-GW						
Diagram	<pre> msgRef --> xs:unsignedByte </pre> <p>Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...</p>						
Type	xs:unsignedByte						
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td> <td style="padding: 2px;">simple</td> </tr> <tr> <td style="padding: 2px;">minOccurs:</td> <td style="padding: 2px;">0</td> </tr> <tr> <td style="padding: 2px;">default:</td> <td style="padding: 2px;">0</td> </tr> </table>	content:	simple	minOccurs:	0	default:	0
content:	simple						
minOccurs:	0						
default:	0						
Source	<xss:element name="msgRef" type="xs:unsignedByte" minOccurs="0" default="0" />						

Element typeSdsSendEvent / sds

Namespace	DR-GW
-----------	-------

Diagram



Type	typeSds
Properties	content: complex
Model	protocol{0,1} , sdsType , msgRef{0,1} , report{0,1} , sdsdata , source{0,1} , target , forward{0,1} , validity{0,1} , tstamp{0,1} , encryption{0,1} , e2eegroup{0,1}
Children	e2eegroup, encryption, forward, msgRef, protocol, report, sdsType, sdsdata, source, target, tstamp, validity
Instance	<pre> <sds xmlns="DR-GW"> <protocol>{0,1}</protocol> <sdsType>{1,1}</sdsType> <msgRef>{0,1}</msgRef> <report>{0,1}</report> <sdsdata>{1,1}</sdsdata> <source>{0,1}</source> <target>{1,1}</target> <forward>{0,1}</forward> <validity>{0,1}</validity> <tstamp>{0,1}</tstamp> <encryption>{0,1}</encryption> <e2eegroup>{0,1}</e2eegroup> </sds> </pre>
Source	<xss:element name="sds" type="typeSds" />

Element interfaceSds / receiveEvent

Namespace	DR-GW
-----------	-------

Diagram	<pre> classDiagram typeEvent < -- typeSdsReceiveEvent typeSdsReceiveEvent "0..1" o--> receiveEvent typeSdsReceiveEvent "0..1" o--> requestId typeSdsReceiveEvent "0..1" o--> result typeSdsReceiveEvent "0..1" o--> sds note "ReceiveEvent is fired upon received SDS." </pre>
Type	typeSdsReceiveEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeSdsReceiveEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , sds
Children	requestId, result, sds
Instance	<pre> <receiveEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <sds>{1,1}</sds> </receiveEvent> </pre>
Source	<code><xss:element name="receiveEvent" type="typeSdsReceiveEvent" /></code>

Element typeSdsReceiveEvent / sds

Namespace	DR-GW
Diagram	<pre> classDiagram typeSds "0..1" o--> sds typeSds "0..1" o--> protocol typeSds "0..1" o--> sdsType typeSds "0..1" o--> msgRef typeSds "0..1" o--> report typeSds "0..1" o--> sdsdata typeSds "0..1" o--> source typeSds "0..1" o--> target typeSds "0..1" o--> forward typeSds "0..1" o--> validity typeSds "0..1" o--> tstamp typeSds "0..1" o--> encryption typeSds "0..1" o--> e2eegroup </pre>

Type	typeSds
Properties	content: complex
Model	protocol{0,1} , sdsType , msgRef{0,1} , report{0,1} , sdsdata , source{0,1} , target , forward{0,1} , validity{0,1} , tstamp{0,1} , encryption{0,1} , e2eegroup{0,1}
Children	e2eegroup, encryption, forward, msgRef, protocol, report, sdsType, sdsdata, source, target, tstamp, validity
Instance	<pre><sds xmlns="DR-GW"> <protocol>{0,1}</protocol> <sdsType>{1,1}</sdsType> <msgRef>{0,1}</msgRef> <report>{0,1}</report> <sdsdata>{1,1}</sdsdata> <source>{0,1}</source> <target>{1,1}</target> <forward>{0,1}</forward> <validity>{0,1}</validity> <tstamp>{0,1}</tstamp> <encryption>{0,1}</encryption> <e2eegroup>{0,1}</e2eegroup> </sds></pre>
Source	<code><xss:element name="sds" type="typeSds" /></code>

Element interfaceSds / reportEvent

Namespace	DR-GW
Diagram	<p>The diagram illustrates the UML class structure for the <code>typeSdsReportEvent</code> class. It extends the <code>typeEvent</code> class, which serves as the extension base. The <code>typeSdsReportEvent</code> class contains the following attributes:</p> <ul style="list-style-type: none"> <code>requestId</code>: multiplicity 0..1 <code>result</code>: multiplicity 0..1 <code>source</code>: multiplicity 0..1 <code>target</code>: multiplicity 0..1 <code>msgRef</code>: multiplicity 0..1 <code>deliveryStatus</code>: multiplicity 0..1 <code>tstamp</code>: multiplicity 0..1 <p>A note at the bottom of the diagram states: "ReportEvent is fired whenever the delivery or consume report is received."</p>
Type	typeSdsReportEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeSdsReportEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , source , target , msgRef , deliveryStatus , tstamp
Children	deliveryStatus, msgRef, requestId, result, source, target, tstamp
Instance	<pre><reportEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <source>{1,1}</source> <target>{1,1}</target> <msgRef>{1,1}</msgRef> <deliveryStatus>{1,1}</deliveryStatus> <tstamp>{1,1}</tstamp> </reportEvent></pre>

Source

```
<xss:element name="reportEvent" type="typeSdsReportEvent" />
```

Element typeSdsReportEvent / source

Namespace	DR-GW
Diagram	<p>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</p>
Type	typeAddress
Properties	content: complex
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<pre><source xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </source></pre>
Source	<pre><xss:element name="source" type="typeAddress" /></pre>

Element typeSdsReportEvent / target

Namespace	DR-GW
-----------	-------

Diagram	<p>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</p>
Type	typeAddress
Properties	content: complex
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<pre><target xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </target></pre>
Source	<code><xss:element name="target" type="typeAddress" /></code>

Element typeSdsReportEvent / msgRef

Namespace	DR-GW
Diagram	<p>Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...</p>
Type	xs:unsignedByte
Properties	content: simple
Source	<code><xss:element name="msgRef" type="xs:unsignedByte" /></code>

Element typeSdsReportEvent / deliveryStatus

Namespace	DR-GW
Diagram	<p>Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...</p>
Type	xs:unsignedByte

Properties	content: simple
Source	<xs:element name="deliveryStatus" type="xs:unsignedByte"/>

Element typeSdsReportEvent / tstamp

Namespace	DR-GW
Diagram	<p>tstamp</p> <p>xs:dateTime</p> <p>Built-in primitive type. The dateTime datatype represents a specific instant of time.</p>
Type	xs:dateTime
Properties	content: simple
Source	<xs:element name="tstamp" type="xs:dateTime"/>

Element drgw / status

Namespace	DR-GW
Diagram	<p>status</p> <p>Type interfaceStatus</p> <p>interfaceStatus</p> <p>send</p> <p>response</p> <p>sendEvent</p> <p>receiveEvent</p> <p>DR-GW-Status element. Use to send/receive status messages. Use only via SOAP.</p>
Type	interfaceStatus
Properties	content: complex
Model	send response sendEvent receiveEvent
Children	receiveEvent, response, send, sendEvent
Instance	<pre><status xmlns="DR-GW"> <send>{1,1}</send> <response>{1,1}</response> <sendEvent>{1,1}</sendEvent> <receiveEvent>{1,1}</receiveEvent> </status></pre>
Source	<xs:element name="status" type="interfaceStatus"/>

Element interfaceStatus / send

Namespace	DR-GW
Diagram	<p>typeStatusSend</p> <p>typeRequest (extension base)</p> <p>requestId</p> <p>status</p>

Type	typeStatusSend
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeStatusSend
Properties	content: complex
Model	requestId , status
Children	requestId, status
Instance	<pre><send xmlns="DR-GW"> <requestId>{1,1}</requestId> <status>{1,1}</status> </send></pre>
Source	<code><xss:element name="send" type="typeStatusSend"/></code>

Element typeStatusSend / status

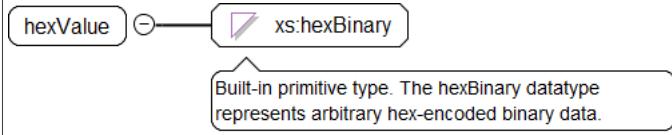
Namespace	DR-GW
Diagram	
Type	typeStatusSend
Properties	content: complex
Model	(value hexValue) , source{0,1} , target , tstamp{0,1}
Children	hexValue, source, target, tstamp, value
Instance	<pre><status xmlns="DR-GW"> <value>{1,1}</value> <hexValue>{1,1}</hexValue> <source>{0,1}</source> <target>{1,1}</target> <tstamp>{0,1}</tstamp> </status></pre>
Source	<code><xss:element name="status" type="typeStatus"/></code>

Element typeStatus / value

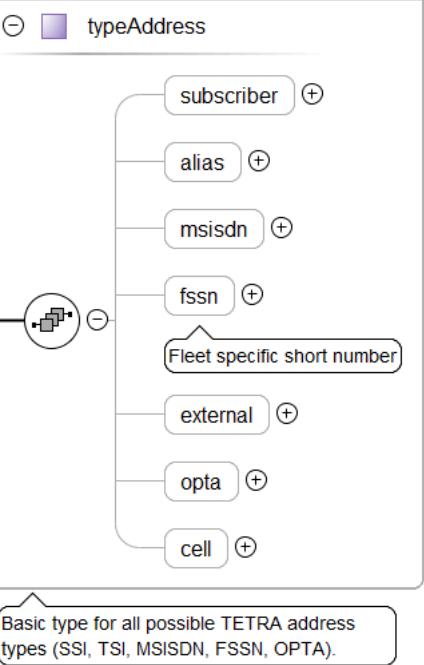
Namespace	DR-GW
Diagram	<p>Built-in derived type. The unsignedShort datatype is derived from unsignedInt by setting the value of maxInclusive to...</p>
Type	xs:unsignedShort
Properties	content: simple
Source	<code><xss:element name="value" type="xs:unsignedShort"/></code>

Element typeStatus / hexValue

Namespace	DR-GW
-----------	-------

Diagram	
Type	xs:hexBinary
Properties	content: simple
Source	<code><xs:element name="hexValue" type="xs:hexBinary" /></code>

Element typeStatus / source

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

Element typeStatus / target

Namespace	DR-GW
-----------	-------

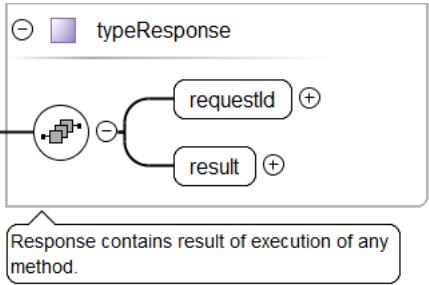
Diagram	<p>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</p>
Type	typeAddress
Properties	content: complex
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<pre><target xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </target></pre>
Source	<code><xss:element name="target" type="typeAddress" /></code>

Element typeStatus / tstamp

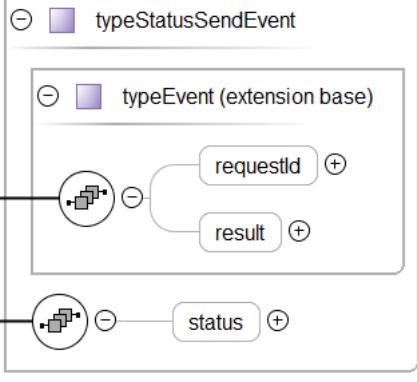
Namespace	DR-GW				
Diagram	<p>Built-in primitive type. The dateTime datatype represents a specific instant of time.</p>				
Type	xs:dateTime				
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td> <td style="padding: 2px;">simple</td> </tr> <tr> <td style="padding: 2px;">minOccurs:</td> <td style="padding: 2px;">0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><xss:element name="tstamp" type="xs:dateTime" minOccurs="0" /></code>				

Element interfaceStatus / response

Namespace	DR-GW
-----------	-------

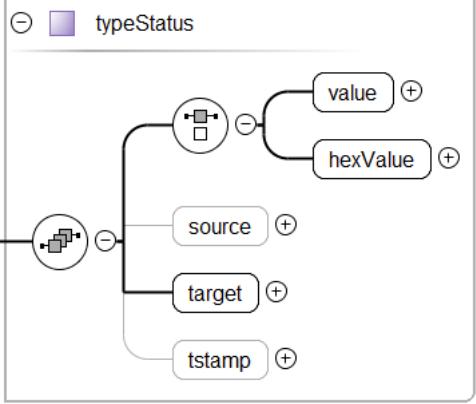
Diagram	 <p>Response contains result of execution of any method.</p>
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<pre><response xmlns="DR-GW"> <requestIds>{1,1}</requestIds> <result>{1,1}</result> </response></pre>
Source	<code><xss:element name="response" type="typeResponse" /></code>

Element interfaceStatus / sendEvent

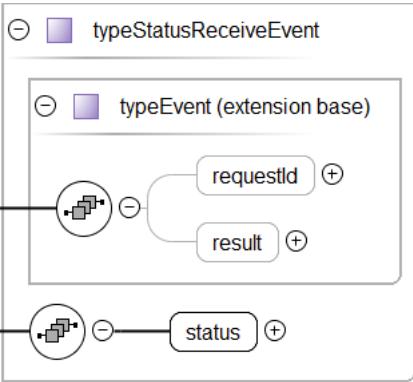
Namespace	DR-GW
Diagram	
Type	typeStatusSendEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeStatusSendEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , status{0,1}
Children	requestId, result, status
Instance	<pre><sendEvent xmlns="DR-GW"> <requestIds>{0,1}</requestIds> <result>{0,1}</result> <status>{0,1}</status> </sendEvent></pre>
Source	<code><xss:element name="sendEvent" type="typeStatusSendEvent" /></code>

Element typeStatusSendEvent / status

Namespace	DR-GW
-----------	-------

Diagram	
Type	typeStatus
Properties	<p>content: complex</p> <p>minOccurs: 0</p>
Model	(value hexValue) , source{0,1} , target , tstamp{0,1}
Children	hexValue, source, target, tstamp, value
Instance	<pre><status xmlns="DR-GW"> <value>{1,1}</value> <hexValue>{1,1}</hexValue> <source>{0,1}</source> <target>{1,1}</target> <tstamp>{0,1}</tstamp> </status></pre>
Source	<code><xss:element name="status" type="typeStatus" minOccurs="0"/></code>

Element interfaceStatus / receiveEvent

Namespace	DR-GW
Diagram	
Type	typeStatusReceiveEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeStatusReceiveEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , status
Children	requestId, result, status
Instance	<pre><receiveEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <status>{1,1}</status> </receiveEvent></pre>
Source	<code><xss:element name="receiveEvent" type="typeStatusReceiveEvent"/></code>

Element typeStatusReceiveEvent / status

Namespace	DR-GW
-----------	-------

Diagram	
Type	typeStatus
Properties	content: complex
Model	(value hexValue) , source{0,1} , target , tstamp{0,1}
Children	hexValue, source, target, tstamp, value
Instance	<pre><status xmlns="DR-GW"> <value>{1,1}</value> <hexValue>{1,1}</hexValue> <source>{0,1}</source> <target>{1,1}</target> <tstamp>{0,1}</tstamp> </status></pre>
Source	<code><xss:element name="status" type="typeStatus"/></code>

Element drgw / org

Namespace	DR-GW
Diagram	
Type	interfaceOrg
Properties	content: complex
Model	get getList response getEvent getListEvent event
Children	event, get, getEvent, getList, getListEvent, response
Instance	<pre><org xmlns="DR-GW"> <get>{1,1}</get> <getList>{1,1}</getList> <response>{1,1}</response> <getEvent>{1,1}</getEvent> <getListEvent>{1,1}</getListEvent> <event>{1,1}</event> </org></pre>

Source

```
<xss:element name="org" type="interfaceOrg" />
```

Element interfaceOrg / get

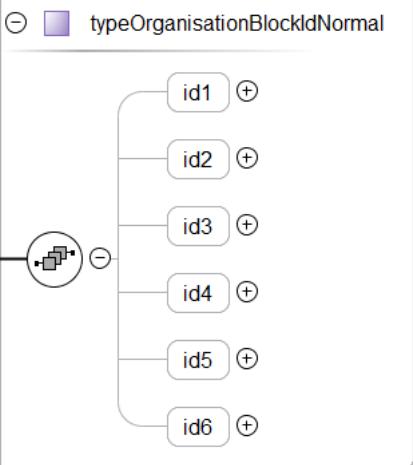
Namespace	DR-GW
Diagram	<pre> classDiagram typeOrgGet "typeOrgGet" typeRequest "typeRequest (extension base)" requestId "requestId" orgblockId "orgblockId" typeOrgGet < -- typeRequest typeRequest --> requestId : 1..1 typeRequest --> orgblockId : 1..1 </pre>
Type	typeOrgGet
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeOrgGet
Properties	content: complex
Model	requestId , orgblockId
Children	orgblockId, requestId
Instance	<pre> <get xmlns="DR-GW"> <requestId>{1,1}</requestId> <orgblockId>{1,1}</orgblockId> </get> </pre>
Source	<pre><xss:element name="get" type="typeOrgGet" /></pre>

Element typeOrgGet / orgblockId

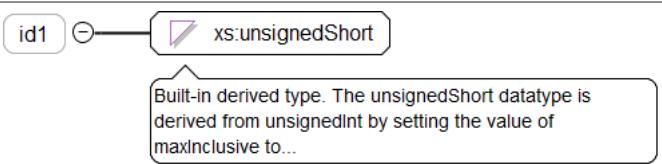
Namespace	DR-GW
Diagram	<pre> classDiagram typeOrganisationBlockId "typeOrganisationBlockId" orgblockId "orgblockId" orgblockIdSimple "orgblockIdSimple" typeOrganisationBlockId < -- orgblockId orgblockId --> orgblockIdSimple : 1..1 orgblockId --> orgblockId : 1..1 </pre>
Type	typeOrganisationBlockId
Properties	content: complex
Model	orgblockId orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<pre> <orgblockId xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <orgblockIdSimple>{1,1}</orgblockIdSimple> </orgblockId> </pre>
Source	<pre><xss:element name="orgblockId" type="typeOrganisationBlockId" /></pre>

Element typeOrganisationBlockId / orgblockId

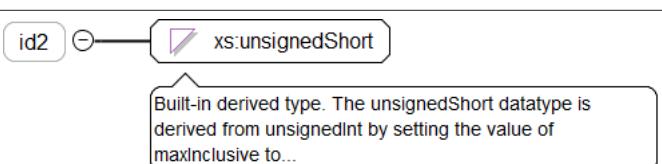
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeOrganisationBlockIdNormal
Properties	content: complex
Model	id1{0,1} , id2{0,1} , id3{0,1} , id4{0,1} , id5{0,1} , id6{0,1}
Children	id1, id2, id3, id4, id5, id6
Instance	<pre><orgblockId xmlns="DR-GW"> <id1>{0,1}</id1> <id2>{0,1}</id2> <id3>{0,1}</id3> <id4>{0,1}</id4> <id5>{0,1}</id5> <id6>{0,1}</id6> </orgblockId></pre>
Source	<code><xs:element name="orgblockId" type="typeOrganisationBlockIdNormal" /></code>

Element typeOrganisationBlockIdNormal / id1

Namespace	DR-GW				
Diagram					
Type	xs:unsignedShort				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><xs:element name="id1" type="xs:unsignedShort" minOccurs="0" /></code>				

Element typeOrganisationBlockIdNormal / id2

Namespace	DR-GW				
Diagram					
Type	xs:unsignedShort				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><xs:element name="id2" type="xs:unsignedShort" minOccurs="0" /></code>				

Element typeOrganisationBlockIdNormal / id3

Namespace	DR-GW				
Diagram	<p>Built-in derived type. The unsignedShort datatype is derived from unsignedInt by setting the value of maxInclusive to...</p>				
Type	xs:unsignedShort				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><xss:element name="id3" type="xs:unsignedShort" minOccurs="0" /></code>				

Element typeOrganisationBlockIdNormal / id4

Namespace	DR-GW				
Diagram	<p>Built-in derived type. The unsignedShort datatype is derived from unsignedInt by setting the value of maxInclusive to...</p>				
Type	xs:unsignedShort				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><xss:element name="id4" type="xs:unsignedShort" minOccurs="0" /></code>				

Element typeOrganisationBlockIdNormal / id5

Namespace	DR-GW				
Diagram	<p>Built-in derived type. The unsignedShort datatype is derived from unsignedInt by setting the value of maxInclusive to...</p>				
Type	xs:unsignedShort				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><xss:element name="id5" type="xs:unsignedShort" minOccurs="0" /></code>				

Element typeOrganisationBlockIdNormal / id6

Namespace	DR-GW				
Diagram	<p>Built-in derived type. The unsignedShort datatype is derived from unsignedInt by setting the value of maxInclusive to...</p>				
Type	xs:unsignedShort				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><xss:element name="id6" type="xs:unsignedShort" minOccurs="0" /></code>				

Element typeOrganisationBlockId / orgblockIdSimple

Namespace	DR-GW
Diagram	<p>The diagram shows a class named "typeOrganisationBlockIdSimple" with a multiplicity of 0..1. A note below it states: "Organisation block send as simple normalized string. The pattern is: id1-id2-id3-id4-id5-id6".</p>
Type	typeOrganisationBlockIdSimple
Properties	content: simple
Facets	<p>pattern</p> <pre>(([0-9] [1-9]\d{0,3} [1-5]\d{4} 6[0-4]\d{3} 65[0-4]\d{2} 655[0-2]\d 6553[0-5])-){0,5}([0-9] [1-9]\d{0,3} [1-5]\d{4} 6[0-4]\d{3} 65[0-4]\d{2} 655[0-2]\d 6553[0-5])</pre>
Source	<code><xss:element name="orgblockIdSimple" type="typeOrganisationBlockIdSimple"/></code>

Element interfaceOrg / getList

Namespace	DR-GW
Diagram	<p>The diagram shows a class named "typeOrgGetList" which extends "typeRequest (extension base)". The "getList" element is shown as a child of "typeOrgGetList" with a multiplicity of 0..1. It has two associations: one to "requestId" and one to "orgblockId".</p>
Type	typeOrgGetList
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeOrgGetList
Properties	content: complex
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId
Instance	<code><getList xmlns="DR-GW"> <requestId>{1,1}</requestId> <orgblockId>{0,1}</orgblockId> </getList></code>
Source	<code><xss:element name="getList" type="typeOrgGetList"/></code>

Element typeOrgGetList / orgblockId

Namespace	DR-GW
Diagram	<p>The diagram shows a class named "typeOrganisationBlockId" which extends "typeOrganisationBlockId". The "orgblockId" element is shown as a child of "typeOrganisationBlockId" with a multiplicity of 0..1. It has two associations: one to "orgblockId" and one to "orgblockIdSimple".</p>
Type	typeOrganisationBlockId

Properties	content: complex minOccurs: 0
Model	orgblockId orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<orgblockId xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <orgblockIdSimple>{1,1}</orgblockIdSimple> </orgblockId>
Source	<xss:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0"/>

Element interfaceOrg / response

Namespace	DR-GW
Diagram	<pre> classDiagram class typeResponse { response * --> requestId 1..1 response * --> result 1..1 } note over requestId, result: Response contains result of execution of any method. </pre>
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<response xmlns="DR-GW"> <requestId>{1,1}</requestId> <result>{1,1}</result> </response>
Source	<xss:element name="response" type="typeResponse" />

Element interfaceOrg / getEvent

Namespace	DR-GW
Diagram	<pre> classDiagram class typeOrgGetEvent { getEvent * --> requestId 1..1 getEvent * --> result 1..1 getEvent * --> orgblock 1..1 } note over typeEvent: typeEvent (extension base) </pre>
Type	typeOrgGetEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeOrgGetEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , orgblock
Children	orgblock, requestId, result
Instance	<getEvent xmlns="DR-GW"> <requestId>{0,1}</requestId>

	<pre><result>{0,1}</result> <orgblock>{1,1}</orgblock> </getEvent></pre>
Source	<code><xs:element name="getEvent" type="typeOrgGetEvent" /></code>

Element typeOrgGetEvent / orgblock

Namespace	DR-GW
Diagram	<pre> classDiagram class typeOrganisationBlock { orgblock* --> orgblockId orgblock* --> alias } orgblockId < -- orgblockIdSimple </pre>
Type	typeOrganisationBlock
Properties	content: complex
Model	orgblockId , alias
Children	alias, orgblockId
Instance	<pre><orgblock xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <alias>{1,1}</alias> </orgblock></pre>
Source	<code><xs:element name="orgblock" type="typeOrganisationBlock" /></code>

Element typeOrganisationBlock / orgblockId

Namespace	DR-GW
Diagram	<pre> classDiagram class typeOrganisationBlockId { orgblockId* --> orgblockId orgblockId* --> orgblockIdSimple } orgblockIdSimple < -- orgblockIdSimple </pre>
Type	typeOrganisationBlockId
Properties	content: complex
Model	orgblockId orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<pre><orgblockId xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <orgblockIdSimple>{1,1}</orgblockIdSimple> </orgblockId></pre>
Source	<code><xs:element name="orgblockId" type="typeOrganisationBlockId" /></code>

Element typeOrganisationBlock / alias

Namespace	DR-GW
Diagram	<p>Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...</p>
Type	xs:normalizedString
Properties	content: simple
Source	<code><xs:element name="alias" type="xs:normalizedString" /></code>

Element interfaceOrg / getListEvent

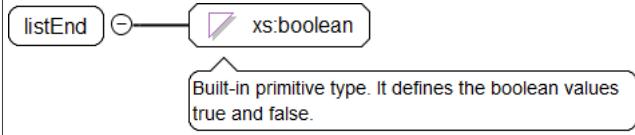
Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeOrgGetListEvent typeEvent { requestId : int? result : int? orgblock : orgblock* listEnd : listEnd } orgblock { orgblockId : string alias : string } listEnd { listEndId : string } </pre>
Type	typeOrgGetListEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeOrgGetListEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , orgblock* , listEnd
Children	listEnd, orgblock, requestId, result
Instance	<getListEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <orgblock>{0,unbounded}</orgblock> <listEnd>{1,1}</listEnd> </getListEvent>
Source	<xss:element name="getListEvent" type="typeOrgGetListEvent" />

Element typeOrgGetListEvent / orgblock

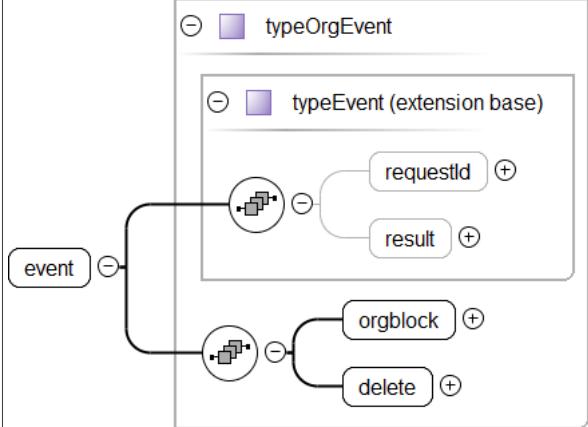
Namespace	DR-GW						
Diagram	<pre> typeOrganisationBlock { orgblockId : string alias : string } </pre>						
Type	typeOrganisationBlock						
Properties	<table> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
content:	complex						
minOccurs:	0						
maxOccurs:	unbounded						
Model	orgblockId , alias						
Children	alias, orgblockId						
Instance	<orgblock xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <alias>{1,1}</alias> </orgblock>						
Source	<xss:element name="orgblock" type="typeOrganisationBlock" minOccurs="0" maxOccurs="unbounded" />						

Element typeOrgGetListEvent / listEnd

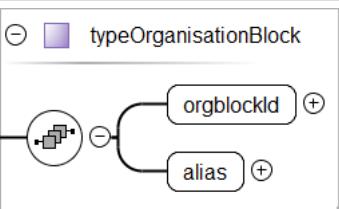
Namespace	DR-GW
-----------	-------

Diagram	 Built-in primitive type. It defines the boolean values true and false.
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="listEnd" type="xs:boolean" />

Element interfaceOrg / event

Namespace	DR-GW
Diagram	
Type	typeOrgEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeOrgEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , orgblock , delete
Children	delete, orgblock, requestId, result
Instance	<pre> <event xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <orgblock>{1,1}</orgblock> <delete>{1,1}</delete> </event> </pre>
Source	<xs:element name="event" type="typeOrgEvent" />

Element typeOrgEvent / orgblock

Namespace	DR-GW
Diagram	
Type	typeOrganisationBlock
Properties	content: complex
Model	orgblockId , alias
Children	alias, orgblockId
Instance	<pre> <orgblock xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> </orgblock> </pre>

	<pre> <alias>{1,1}</alias> </orgblock> </pre>
Source	<pre><xss:element name="orgblock" type="typeOrganisationBlock"/></pre>

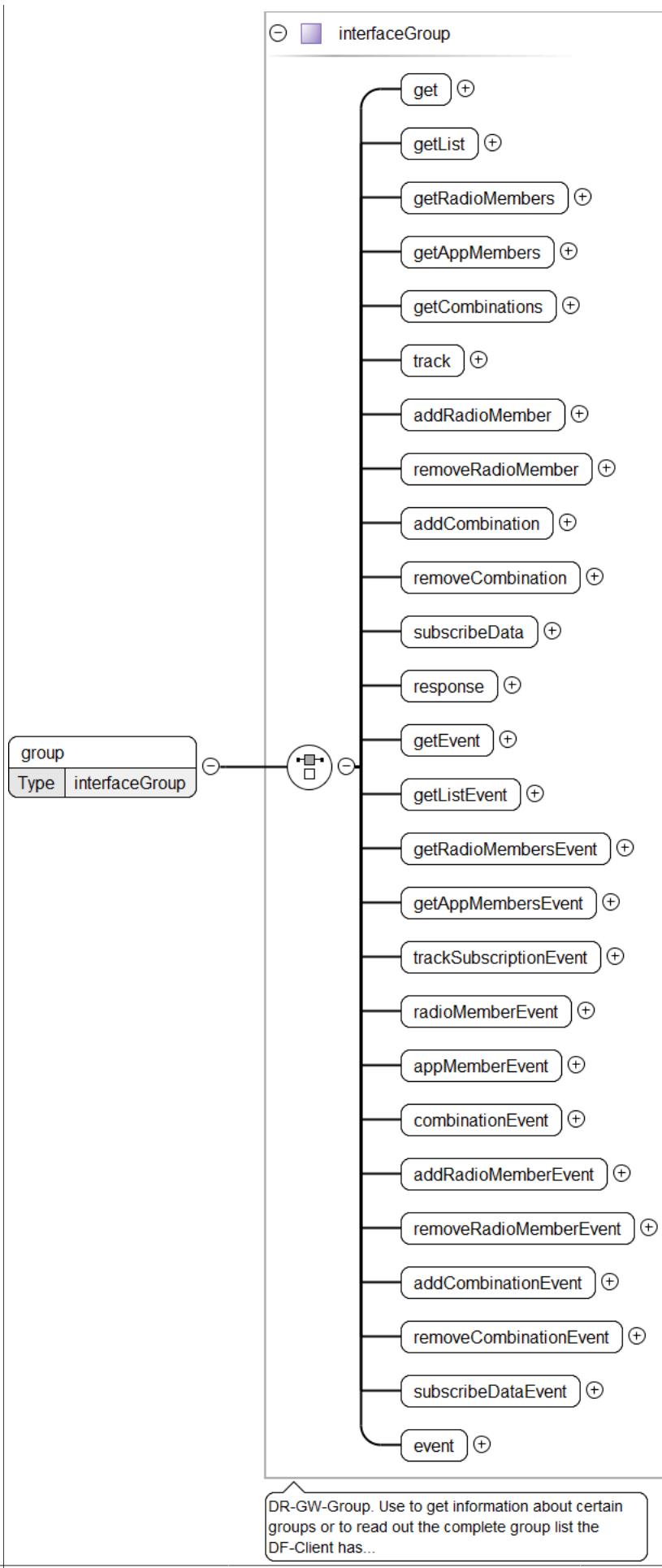
Element typeOrgEvent / delete

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<pre><xss:element name="delete" type="xs:boolean"/></pre>

Element drgw / group

Namespace	DR-GW
-----------	-------

Diagram



Type	interfaceGroup
Properties	content: complex
Model	get getList getRadioMembers getAppMembers getCombinations track addRadioMember removeRadioMember addCombination removeCombination subscribeData response getEvent getListEvent getRadioMembersEvent getAppMembersEvent trackSubscriptionEvent radioMemberEvent appMemberEvent combinationEvent addRadioMemberEvent removeRadioMemberEvent addCombinationEvent removeCombinationEvent subscribeDataEvent event
Children	addCombination, addCombinationEvent, addRadioMember, addRadioMemberEvent, appMemberEvent, combinationEvent, event, get, getAppMembers, getAppMembersEvent, getCombinations, getEvent, getList, getListEvent, getRadioMembers, getRadioMembersEvent, radioMemberEvent, removeCombination, removeCombinationEvent, removeRadioMember, removeRadioMemberEvent, response, subscribeData, subscribeDataEvent, track, trackSubscriptionEvent
Instance	<pre> <group xmlns="DR-GW"> <get>{1,1}</get> <getList>{1,1}</getList> <getRadioMembers>{1,1}</getRadioMembers> < getAppMembers>{1,1}</ getAppMembers> <getCombinations>{1,1}</getCombinations> <track>{1,1}</track> <addRadioMember>{1,1}</addRadioMember> <removeRadioMember>{1,1}</removeRadioMember> <addCombination>{1,1}</addCombination> <removeCombination>{1,1}</removeCombination> <subscribeData>{1,1}</subscribeData> <response>{1,1}</response> <getEvent>{1,1}</getEvent> <getListEvent>{1,1}</getListEvent> <getRadioMembersEvent>{1,1}</getRadioMembersEvent> < getAppMembersEvent>{1,1}</ getAppMembersEvent> <trackSubscriptionEvent>{1,1}</trackSubscriptionEvent> <radioMemberEvent>{1,1}</radioMemberEvent> <appMemberEvent>{1,1}</appMemberEvent> <combinationEvent>{1,1}</combinationEvent> <addRadioMemberEvent>{1,1}</addRadioMemberEvent> <removeRadioMemberEvent>{1,1}</removeRadioMemberEvent> <addCombinationEvent>{1,1}</addCombinationEvent> <removeCombinationEvent>{1,1}</removeCombinationEvent> <subscribeDataEvent>{1,1}</subscribeDataEvent> <event>{1,1}</event> </group></pre>
Source	<xss:element name="group" type="interfaceGroup" />

Element interfaceGroup / get

Namespace	DR-GW
Diagram	<pre> classDiagram class typeGroupGet { <<interface group>> <<extension base="typeRequest">> <<operations>> <<get>> : <<requestId>> <<group>> <<group>> : <<group>> </operations> } </pre>
Type	typeGroupGet
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupGet
Properties	content: complex
Model	requestId , group
Children	group, requestId
Instance	<pre> <get xmlns="DR-GW"> <requestId>{1,1}</requestId> <group>{1,1}</group> </get></pre>
Source	<xss:element name="get" type="typeGroupGet" />

Element typeGroupGet / group

Namespace	DR-GW
Diagram	<pre> classDiagram typeGroupGet "group" --> typeSubscriberAddress typeSubscriberAddress "ssi" +--> ssi typeSubscriberAddress "tsi" +--> tsi </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre> <group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group> </pre>
Source	<xss:element name="group" type="typeSubscriberAddress"/>

Element interfaceGroup / getList

Namespace	DR-GW
Diagram	<pre> classDiagram interfaceGroup "getList" --> typeGroupGetList typeGroupGetList "requestId" +--> requestId typeGroupGetList "orgblockId" +--> orgblockId </pre>
Type	typeGroupGetList
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupGetList
Properties	content: complex
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId
Instance	<pre> <getList xmlns="DR-GW"> <requestId>{1,1}</requestId> <orgblockId>{0,1}</orgblockId> </getList> </pre>
Source	<xss:element name="getList" type="typeGroupGetList"/>

Element typeGroupGetList / orgblockID

Namespace	DR-GW
Diagram	<pre> classDiagram typeGroupGetList "orgblockID" --> typeOrganisationBlockId typeOrganisationBlockId "orgblockId" +--> orgblockId typeOrganisationBlockId "orgblockIdSimple" +--> orgblockIdSimple </pre>

Type	typeOrganisationBlockId
Properties	content: complex minOccurs: 0
Model	orgblockId orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<pre><orgblockId xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <orgblockIdSimple>{1,1}</orgblockIdSimple> </orgblockId></pre>
Source	<code><xss:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0" /></code>

Element interfaceGroup / getRadioMembers

Namespace	DR-GW
Diagram	<pre> classDiagram typeRequest < -- typeGroupGetRadioMembers typeGroupGetRadioMembers "1..1" -- "1..1" requestId typeGroupGetRadioMembers "1..1" -- "1..1" group </pre>
Type	typeGroupGetRadioMembers
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupGetRadioMembers
Properties	content: complex
Model	requestId , group
Children	group, requestId
Instance	<pre><getRadioMembers xmlns="DR-GW"> <requestId>{1,1}</requestId> <group>{1,1}</group> </getRadioMembers></pre>
Source	<code><xss:element name="getRadioMembers" type="typeGroupGetRadioMembers" /></code>

Element typeGroupGetRadioMembers / group

Namespace	DR-GW
Diagram	<pre> classDiagram typeSubscriberAddress "1..1" -- "1..1" group group "1..1" -- "1..1" ssi group "1..1" -- "1..1" tsi </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group></pre>
Source	<code><xss:element name="group" type="typeSubscriberAddress" /></code>

Element interfaceGroup / getAppMembers

Namespace	DR-GW
Diagram	<pre> classDiagram typeGroupGetAppMembers < -- typeRequest typeGroupGetAppMembers "1..1" --> requestId : String typeGroupGetAppMembers "0..1" --> group : String </pre>
Type	typeGroupGetAppMembers
Type hierarchy	<ul style="list-style-type: none"> • typeRequest <ul style="list-style-type: none"> • typeGroupGetAppMembers
Properties	content: complex
Model	requestId , group
Children	group, requestId
Instance	<pre> <getAppMembers xmlns="DR-GW"> <requestId>{1,1}</requestId> <group>{1,1}</group> </getAppMembers> </pre>
Source	<code><xss:element name="getAppMembers" type="typeGroupGetAppMembers" /></code>

Element typeGroupGetAppMembers / group

Namespace	DR-GW
Diagram	<pre> classDiagram typeSubscriberAddress < -- typeGroup typeSubscriberAddress "1..1" --> ssi : String typeSubscriberAddress "1..1" --> tsi : String </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre> <group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group> </pre>
Source	<code><xss:element name="group" type="typeSubscriberAddress" /></code>

Element interfaceGroup / getCombinations

Namespace	DR-GW
-----------	-------

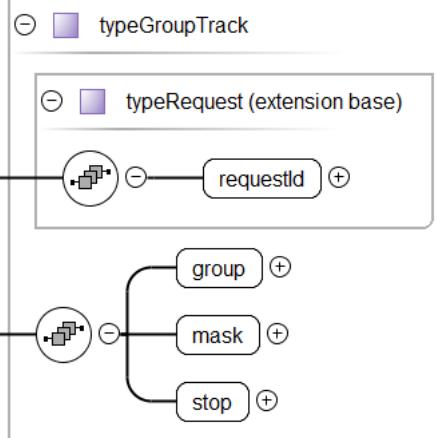
Diagram	<pre> classDiagram typeGroupGetCombinations < -- typeRequest typeGroupGetCombinations "0..1" --> "1..1" requestId typeGroupGetCombinations "0..1" --> "1..1" group note over typeGroupGetCombinations: The method requests the groups that belong to the same combined group as the group specified. </pre>
Type	typeGroupGetCombinations
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupGetCombinations
Properties	content: complex
Model	requestId , group
Children	group, requestId
Instance	<pre> <getCombinations xmlns="DR-GW"> <requestId>{1,1}</requestId> <group>{1,1}</group> </getCombinations> </pre>
Source	<code><x:element name="getCombinations" type="typeGroupGetCombinations" /></code>

Element typeGroupGetCombinations / group

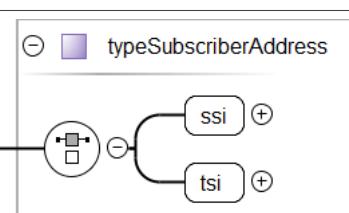
Namespace	DR-GW
Diagram	<pre> classDiagram typeSubscriberAddress "0..1" --> "1..1" ssi typeSubscriberAddress "0..1" --> "1..1" tsi note over typeSubscriberAddress: The method requests the groups that belong to the same combined group as the group specified. </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre> <group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group> </pre>
Source	<code><x:element name="group" type="typeSubscriberAddress" /></code>

Element interfaceGroup / track

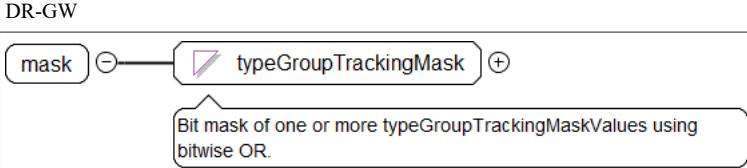
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeGroupTrack
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupTrack
Properties	content: complex
Model	requestId , group , mask , stop
Children	group, mask, requestId, stop
Instance	<pre><track xmlns="DR-GW"> <requestId>{1,1}</requestId> <group>{1,1}</group> <mask>{1,1}</mask> <stop>{1,1}</stop> </track></pre>
Source	<code><xss:element name="track" type="typeGroupTrack" /></code>

Element typeGroupTrack / group

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group></pre>
Source	<code><xss:element name="group" type="typeSubscriberAddress" /></code>

Element typeGroupTrack / mask

Namespace	DR-GW
Diagram	
Type	typeGroupTrackingMask

Properties	content: simple
Source	<xs:element name="mask" type="typeGroupTrackingMask"/>

Element typeGroupTrack / stop

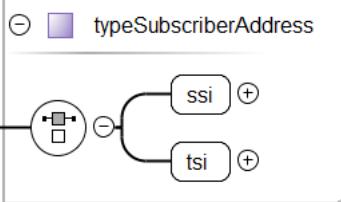
Namespace	DR-GW
Diagram	<p>The diagram shows a class named 'stop' with a multiplicity of 0..1. It has a directed association labeled 'xs:boolean' with a target compartment containing the icon for the xs:boolean primitive type. A callout box states: 'Built-in primitive type. It defines the boolean values true and false.'</p>
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="stop" type="xs:boolean"/>

Element interfaceGroup / addRadioMember

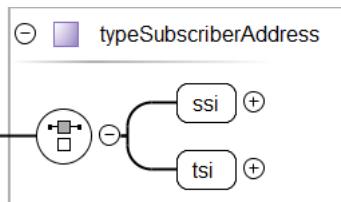
Namespace	DR-GW
Diagram	<p>The diagram shows a class named 'addRadioMember' with a multiplicity of 0..1. It has a directed association labeled 'typeRequest (extension base)' with a target compartment containing the 'typeRequest' class. This association is marked with a circled plus sign. Below it, there are three more associations: one to 'requestId' (multiplicity 0..1), one to 'radio' (multiplicity 1..1), one to 'group' (multiplicity 1..1), and one to 'membership' (multiplicity 0..1). Each of these last three associations is marked with a circled plus sign. A callout box states: 'Requests the addition of a radio subscriber to a group. This might cause DGNA operation in the air interface.'</p>
Type	typeGroupAddRadioMember
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupAddRadioMember
Properties	content: complex
Model	requestId , radio , group , membership{0,1}
Children	group, membership, radio, requestId
Instance	<pre><addRadioMember xmlns="DR-GW"> <requestId>{1,1}</requestId> <radio>{1,1}</radio> <group>{1,1}</group> <membership>{0,1}</membership> </addRadioMember></pre>
Source	<xs:element name="addRadioMember" type="typeGroupAddRadioMember"/>

Element typeGroupAddRadioMember / radio

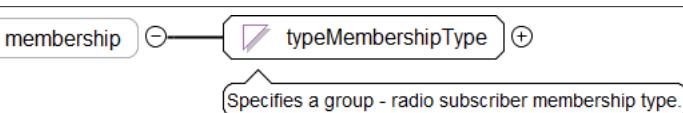
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio></pre>
Source	<code><xss:element name="radio" type="typeSubscriberAddress"/></code>

Element typeGroupAddRadioMember / group

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group></pre>
Source	<code><xss:element name="group" type="typeSubscriberAddress"/></code>

Element typeGroupAddRadioMember / membership

Namespace	DR-GW
Diagram	 Specifies a group - radio subscriber membership type.
Type	typeMembershipType
Properties	content: simple minOccurs: 0
Facets	enumeration unknown enumeration permanent enumeration visiting
Source	<code><xss:element name="membership" type="typeMembershipType" minOccurs="0" /></code>

Element interfaceGroup / removeRadioMember

Namespace	DR-GW
-----------	-------

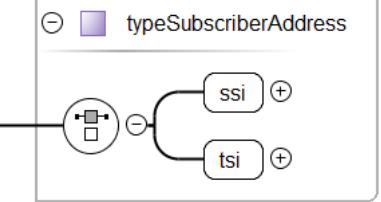
Diagram	<pre> classDiagram typeGroupRemoveRadioMember < -- typeRequest typeGroupRemoveRadioMember { <> requestId : <<radio>> <> radio : <<radio>> <> group : <<radio>> } </pre> <p>Requests removing a radio subscriber from a group. This might cause DGNA operation in the air interface.</p>
Type	typeGroupRemoveRadioMember
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupRemoveRadioMember
Properties	content: complex
Model	requestId , radio , group
Children	group, radio, requestId
Instance	<removeRadioMember xmlns="DR-GW"> <requestId>{1,1}</requestId> <radio>{1,1}</radio> <group>{1,1}</group> </removeRadioMember>
Source	<xss:element name="removeRadioMember" type="typeGroupRemoveRadioMember"/>

Element typeGroupRemoveRadioMember / radio

Namespace	DR-GW
Diagram	<pre> classDiagram typeSubscriberAddress { <> radio : <<radio>> <> ssi : <<radio>> <> tsi : <<radio>> } </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xss:element name="radio" type="typeSubscriberAddress"/>

Element typeGroupRemoveRadioMember / group

Namespace	DR-GW
-----------	-------

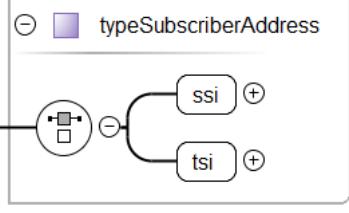
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xss:element name="group" type="typeSubscriberAddress"/>

Element interfaceGroup / addCombination

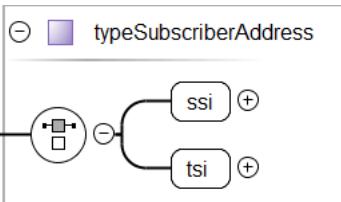
Namespace	DR-GW
Diagram	<p>Requests the addition of a group to a combined group.</p>
Type	typeGroupAddCombination
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupAddCombination
Properties	content: complex
Model	requestId , group , baseGroup , force{0,1}
Children	baseGroup, force, group, requestId
Instance	<addCombination xmlns="DR-GW"> <requestId>{1,1}</requestId> <group>{1,1}</group> <baseGroup>{1,1}</baseGroup> <force>{0,1}</force> </addCombination>
Source	<xss:element name="addCombination" type="typeGroupAddCombination"/>

Element typeGroupAddCombination / group

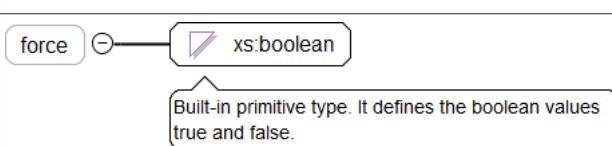
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group></pre>
Source	<code><xss:element name="group" type="typeSubscriberAddress"/></code>

Element typeGroupAddCombination / baseGroup

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><baseGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </baseGroup></pre>
Source	<code><xss:element name="baseGroup" type="typeSubscriberAddress"/></code>

Element typeGroupAddCombination / force

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>default: true</p>
Source	<code><xss:element name="force" type="xs:boolean" minOccurs="0" default="true"/></code>

Element interfaceGroup / removeCombination

Namespace	DR-GW
-----------	-------

Diagram	<pre> classDiagram class removeCombination { <<removeCombination>> } class typeRequest { <<typeRequest (extension base)>> } removeCombination "1..1" --> "1..1" typeRequest : requestId removeCombination "1..1" --> "1..1" typeRequest : group removeCombination "1..1" --> "1..1" typeRequest : baseGroup typeRequest "1..1" --> "1..1" typeGroupRemoveCombination : <<extension base>> note over typeGroupRemoveCombination: Requests removing a group from a combined group. </pre>
Type	typeGroupRemoveCombination
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupRemoveCombination
Properties	content: complex
Model	requestId , group , baseGroup
Children	baseGroup, group, requestId
Instance	<pre> <removeCombination xmlns="DR-GW"> <requestId>{1,1}</requestId> <group>{1,1}</group> <baseGroup>{1,1}</baseGroup> </removeCombination> </pre>
Source	<code><xss:element name="removeCombination" type="typeGroupRemoveCombination"/></code>

Element typeGroupRemoveCombination / group

Namespace	DR-GW
Diagram	<pre> classDiagram class group { <<group>> } group "1..1" --> "1..1" typeSubscriberAddress : ssi group "1..1" --> "1..1" typeSubscriberAddress : tsi </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre> <group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group> </pre>
Source	<code><xss:element name="group" type="typeSubscriberAddress"/></code>

Element typeGroupRemoveCombination / baseGroup

Namespace	DR-GW
Diagram	<pre> classDiagram class baseGroup { <<baseGroup>> } baseGroup "1..1" --> "1..1" typeSubscriberAddress : ssi baseGroup "1..1" --> "1..1" typeSubscriberAddress : tsi </pre>

Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<baseGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </baseGroup>
Source	<xss:element name="baseGroup" type="typeSubscriberAddress" />

Element interfaceGroup / subscribeData

Namespace	DR-GW
Diagram	<pre> classDiagram typeRequest < -- typeGroupSubscribeData typeRequest { -requestId -group } typeGroupSubscribeData { +subscribeData } typeGroupSubscribeData < -- typeRequest typeGroupSubscribeData { +group } </pre>
Type	typeGroupSubscribeData
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupSubscribeData
Properties	content: complex
Model	requestId , group+
Children	group, requestId
Instance	<subscribeData xmlns="DR-GW"> <requestId>{1,1}</requestId> <group>{1,unbounded}</group> </subscribeData>
Source	<xss:element name="subscribeData" type="typeGroupSubscribeData" />

Element typeGroupSubscribeData / group

Namespace	DR-GW				
Diagram	<pre> classDiagram typeGroupDataSubscription < -- group typeGroupDataSubscription { -addr -useSDS -useStatus } group { +useSDS } group < -- typeGroupDataSubscription </pre>				
Type	typeGroupDataSubscription				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	maxOccurs:	unbounded
content:	complex				
maxOccurs:	unbounded				
Model	addr , useSDS , useStatus				
Children	addr, useSDS, useStatus				
Instance	<group xmlns="DR-GW"> <addr>{1,1}</addr> <useSDS>{1,1}</useSDS> <useStatus>{1,1}</useStatus> </group>				

Source

```
<xss:element name="group" type="typeGroupDataSubscription" maxOccurs="unbounded"/>
```

Element typeGroupDataSubscription / addr

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><addr xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </addr></pre>
Source	<pre><xss:element name="addr" type="typeSubscriberAddress" /></pre>

Element typeGroupDataSubscription / useSDS

Namespace	DR-GW
Diagram	<p>Built-in primitive type. It defines the boolean values true and false.</p>
Type	xs:boolean
Properties	content: simple
Source	<pre><xss:element name="useSDS" type="xs:boolean" /></pre>

Element typeGroupDataSubscription / useStatus

Namespace	DR-GW
Diagram	<p>Built-in primitive type. It defines the boolean values true and false.</p>
Type	xs:boolean
Properties	content: simple
Source	<pre><xss:element name="useStatus" type="xs:boolean" /></pre>

Element interfaceGroup / response

Namespace	DR-GW
Diagram	<p>Response contains result of execution of any method.</p>

Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<response xmlns="DR-GW"> <requestId>{1,1}</requestId> <result>{1,1}</result> </response>
Source	<xss:element name="response" type="typeResponse" />

Element interfaceGroup / getEvent

Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeGroupGetEvent typeGroupGetEvent "1..1" -- "1..1" getEvent typeGroupGetEvent "1..1" -- "1..1" requestId typeGroupGetEvent "1..1" -- "1..1" result typeGroupGetEvent "1..1" -- "1..1" group </pre>
Type	typeGroupGetEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupGetEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group
Children	group, requestId, result
Instance	<getEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> </getEvent>
Source	<xss:element name="getEvent" type="typeGroupGetEvent" />

Element typeGroupGetEvent / group

Namespace	DR-GW
Diagram	<pre> typeGroup "1..1" -- "1..1" group typeGroup "1..1" -- "1..1" addr typeGroup "1..1" -- "1..1" alias typeGroup "1..1" -- "1..1" orgblockId </pre>
Type	typeGroup
Properties	content: complex
Model	addr , alias , orgblockId
Children	addr, alias, orgblockId
Instance	<group xmlns="DR-GW"> <addr>{1,1}</addr>

	<pre> <alias>{1,1}</alias> <orgblockId>{1,1}</orgblockId> </group> </pre>
Source	<pre><xs:element name="group" type="typeGroup" /></pre>

Element typeGroup / addr

Namespace	DR-GW
Diagram	<pre> classDiagram typeSubscriberAddress { -addr } typeSubscriberAddress < -- typeGroup typeGroup --> ssi typeGroup --> tsi </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre> <addr xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </addr> </pre>
Source	<pre><xs:element name="addr" type="typeSubscriberAddress" /></pre>

Element typeGroup / alias

Namespace	DR-GW
Diagram	<pre> alias <--> xs:normalizedString </pre> <p>Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...</p>
Type	xs:normalizedString
Properties	content: simple
Source	<pre><xs:element name="alias" type="xs:normalizedString" /></pre>

Element typeGroup / orgblockId

Namespace	DR-GW
Diagram	<pre> typeOrganisationBlockId { -orgblockId } typeOrganisationBlockId < -- typeGroup typeGroup --> orgblockId typeGroup --> orgblockIdSimple </pre>
Type	typeOrganisationBlockId
Properties	content: complex
Model	orgblockId orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<pre> <orgblockId xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <orgblockIdSimple>{1,1}</orgblockIdSimple> </orgblockId> </pre>
Source	<pre><xs:element name="orgblockId" type="typeOrganisationBlockId" /></pre>

Element interfaceGroup / getListEvent

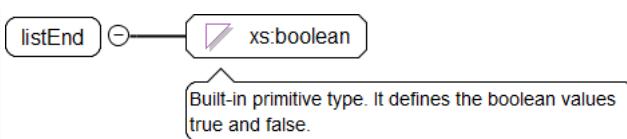
Namespace	DR-GW
Diagram	<pre> classDiagram typeGroupGetListEvent < -- typeEvent typeEvent "0..∞" --> listEnd : group typeEvent "0..∞" --> requestId : requestId typeEvent "0..∞" --> result : result getListEvent --> typeEvent : getListEvent </pre>
Type	typeGroupGetListEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupGetListEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group* , listEnd
Children	group, listEnd, requestId, result
Instance	<pre> <getListEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{0,unbounded}</group> <listEnd>{1,1}</listEnd> </getListEvent> </pre>
Source	<code><xss:element name="getListEvent" type="typeGroupGetListEvent" /></code>

Element typeGroupGetListEvent / group

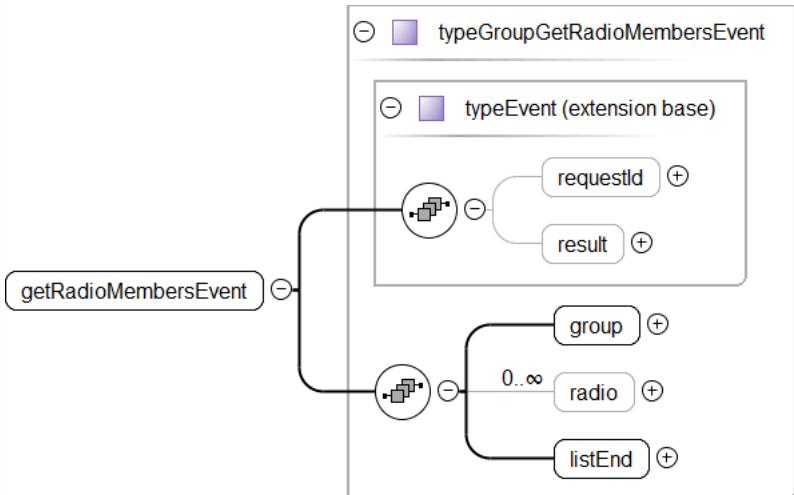
Namespace	DR-GW
Diagram	<pre> classDiagram typeGroupGetListEvent < -- typeGroup typeGroup < -- typeGroupGetListEvent typeGroup < -- typeEvent typeGroup "0..∞" --> addr : addr typeGroup "0..∞" --> alias : alias typeGroup "0..∞" --> orgblockId : orgblockId group --> typeGroup : group </pre>
Type	typeGroup
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Model	addr , alias , orgblockId
Children	addr, alias, orgblockId
Instance	<pre> <group xmlns="DR-GW"> <addr>{1,1}</addr> <alias>{1,1}</alias> <orgblockId>{1,1}</orgblockId> </group> </pre>
Source	<code><xss:element name="group" type="typeGroup" minOccurs="0" maxOccurs="unbounded" /></code>

Element typeGroupGetListEvent / listEnd

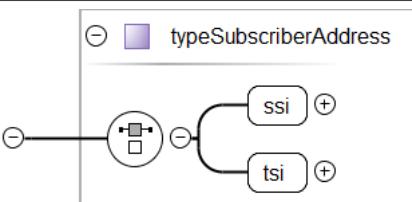
Namespace	DR-GW
-----------	-------

Diagram	
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="listEnd" type="xs:boolean" />

Element interfaceGroup / getRadioMembersEvent

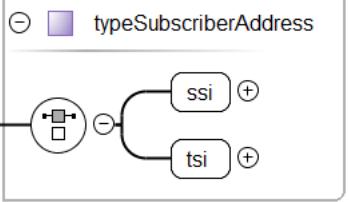
Namespace	DR-GW
Diagram	
Type	typeGroupGetRadioMembersEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupGetRadioMembersEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , radio* , listEnd
Children	group, listEnd, radio, requestId, result
Instance	<pre><getRadioMembersEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> <radio>{0,unbounded}</radio> <listEnd>{1,1}</listEnd> </getRadioMembersEvent></pre>
Source	<xs:element name="getRadioMembersEvent" type="typeGroupGetRadioMembersEvent" />

Element typeGroupGetRadioMembersEvent / group

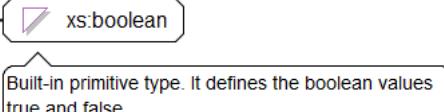
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi

Instance	<pre><group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group></pre>
Source	<pre><xss:element name="group" type="typeSubscriberAddress" /></pre>

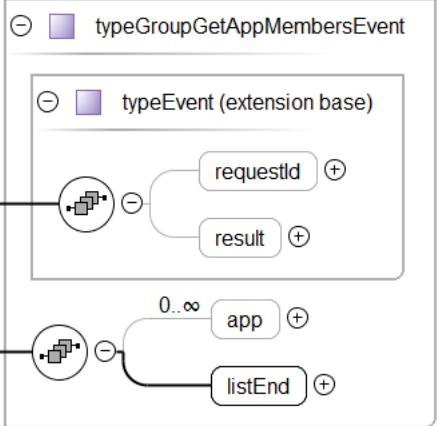
Element typeGroupGetRadioMembersEvent / radio

Namespace	DR-GW						
Diagram							
Type	typeSubscriberAddress						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
content:	complex						
minOccurs:	0						
maxOccurs:	unbounded						
Model	ssi tsi						
Children	ssi, tsi						
Instance	<pre><radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio></pre>						
Source	<pre><xss:element name="radio" type="typeSubscriberAddress" minOccurs="0" maxOccurs="unbounded" /></pre>						

Element typeGroupGetRadioMembersEvent / listEnd

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<pre><xss:element name="listEnd" type="xs:boolean" /></pre>

Element interfaceGroup / getAppMembersEvent

Namespace	DR-GW
Diagram	
Type	typeGroupGetAppMembersEvent

Type hierarchy	<ul style="list-style-type: none"> • typeEvent <ul style="list-style-type: none"> • typeGroupGetAppMembersEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , app* , listEnd
Children	app, listEnd, requestId, result
Instance	<pre><getAppMembersEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <app>{0,unbounded}</app> <listEnd>{1,1}</listEnd> </getAppMembersEvent></pre>
Source	<code><xss:element name="getAppMembersEvent" type="typeGroupGetAppMembersEvent" /></code>

Element typeGroupGetAppMembersEvent / app

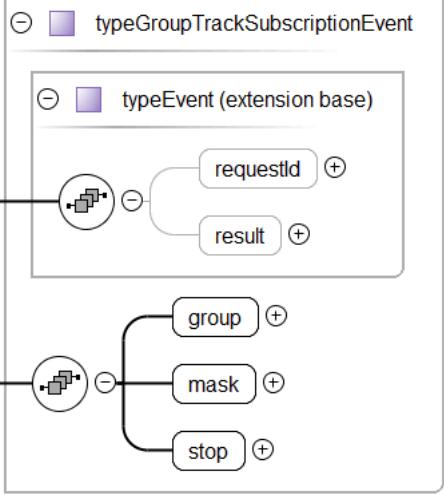
Namespace	DR-GW						
Diagram							
Type	typeSubscriberAddress						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
content:	complex						
minOccurs:	0						
maxOccurs:	unbounded						
Model	ssi tsi						
Children	ssi, tsi						
Instance	<pre><app xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </app></pre>						
Source	<code><xss:element name="app" type="typeSubscriberAddress" minOccurs="0" maxOccurs="unbounded" /></code>						

Element typeGroupGetAppMembersEvent / listEnd

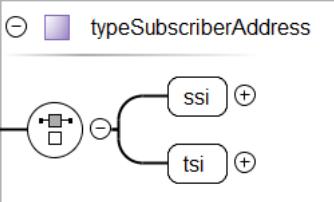
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<code><xss:element name="listEnd" type="xs:boolean" /></code>

Element interfaceGroup / trackSubscriptionEvent

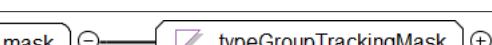
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeGroupTrackSubscriptionEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupTrackSubscriptionEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , mask , stop
Children	group, mask, requestId, result, stop
Instance	<pre><trackSubscriptionEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> <mask>{1,1}</mask> <stop>{1,1}</stop> </trackSubscriptionEvent></pre>
Source	<code><xss:element name="trackSubscriptionEvent" type="typeGroupTrackSubscriptionEvent" /></code>

Element typeGroupTrackSubscriptionEvent / group

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group></pre>
Source	<code><xss:element name="group" type="typeSubscriberAddress" /></code>

Element typeGroupTrackSubscriptionEvent / mask

Namespace	DR-GW
Diagram	 <p>Bit mask of one or more typeGroupTrackingMaskValues using bitwise OR.</p>

Type	typeGroupTrackingMask
Properties	content: simple
Source	<xs:element name="mask" type="typeGroupTrackingMask" />

Element typeGroupTrackSubscriptionEvent / stop

Namespace	DR-GW
Diagram	<p>The diagram shows a class named 'stop' with a multiplicity of 0..1. It has a directed association labeled '-' pointing to a class named 'xs:boolean'. A callout box points to the 'xs:boolean' class with the text: 'Built-in primitive type. It defines the boolean values true and false.'</p>
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="stop" type="xs:boolean" />

Element interfaceGroup / radioMemberEvent

Namespace	DR-GW
Diagram	<p>The diagram shows a class named 'radioMemberEvent' with a multiplicity of 0..1. It has a directed association labeled '-' pointing to a class named 'typeEvent' (extension base). The 'typeEvent' class has two associations: one labeled 'requestId' with multiplicity 0..1 and another labeled 'result' with multiplicity 0..1. Another association labeled '-' points to a class with three children: 'group' (multiplicity 0..1), 'radio' (multiplicity 0..1), and 'delete' (multiplicity 0..1).</p>
Type	typeGroupRadioMemberEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupRadioMemberEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , radio , delete
Children	delete, group, radio, requestId, result
Instance	<pre> <radioMemberEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> <radio>{1,1}</radio> <delete>{1,1}</delete> </radioMemberEvent> </pre>
Source	<xs:element name="radioMemberEvent" type="typeGroupRadioMemberEvent" />

Element typeGroupRadioMemberEvent / group

Namespace	DR-GW
-----------	-------

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group></pre>
Source	<code><xss:element name="group" type="typeSubscriberAddress"/></code>

Element typeGroupRadioMemberEvent / radio

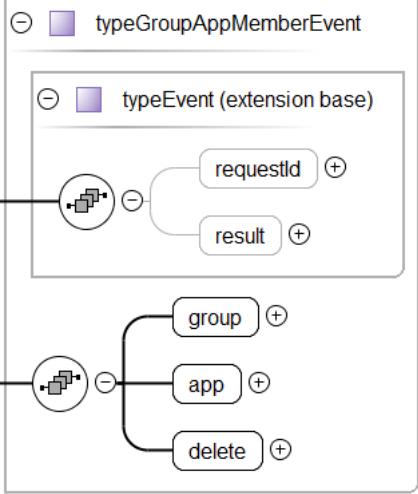
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio></pre>
Source	<code><xss:element name="radio" type="typeSubscriberAddress"/></code>

Element typeGroupRadioMemberEvent / delete

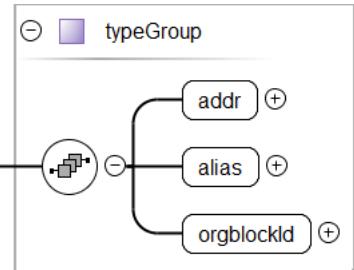
Namespace	DR-GW
Diagram	 A callout box points to the xs:boolean element with the text: "Built-in primitive type. It defines the boolean values true and false.".
Type	xs:boolean
Properties	content: simple
Source	<code><xss:element name="delete" type="xs:boolean"/></code>

Element interfaceGroup / appMemberEvent

Namespace	DR-GW
-----------	-------

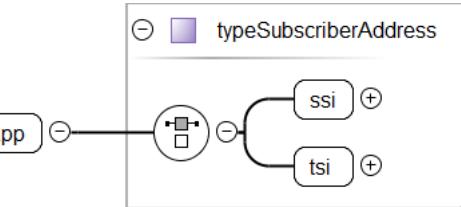
Diagram	
Type	typeGroupAppMemberEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupAppMemberEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , app , delete
Children	app, delete, group, requestId, result
Instance	<pre><appMemberEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> <app>{1,1}</app> <delete>{1,1}</delete> </appMemberEvent></pre>
Source	<code><xs:element name="appMemberEvent" type="typeGroupAppMemberEvent" /></code>

Element typeGroupAppMemberEvent / group

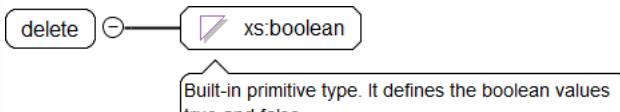
Namespace	DR-GW
Diagram	
Type	typeGroup
Properties	content: complex
Model	addr , alias , orgblockId
Children	addr, alias, orgblockId
Instance	<pre><group xmlns="DR-GW"> <addr>{1,1}</addr> <alias>{1,1}</alias> <orgblockId>{1,1}</orgblockId> </group></pre>
Source	<code><xs:element name="group" type="typeGroup" /></code>

Element typeGroupAppMemberEvent / app

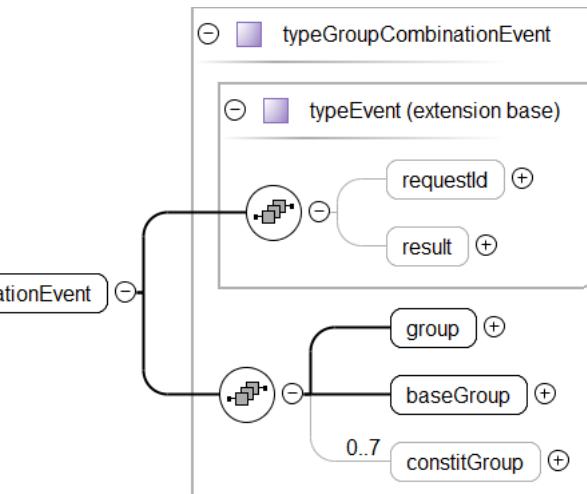
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<app xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </app>
Source	<xs:element name="app" type="typeSubscriberAddress"/>

Element typeGroupAppMemberEvent / delete

Namespace	DR-GW
Diagram	 A callout box points to xs:boolean with the text: Built-in primitive type. It defines the boolean values true and false.
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="delete" type="xs:boolean"/>

Element interfaceGroup / combinationEvent

Namespace	DR-GW
Diagram	
Type	typeGroupCombinationEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupCombinationEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , baseGroup , constitGroup{0,7}
Children	baseGroup, constitGroup, group, requestId, result
Instance	<combinationEvent xmlns="DR-GW">

	<pre> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> <baseGroup>{1,1}</baseGroup> <constitGroup>{0,7}</constitGroup> </combinationEvent> </pre>
Source	<xss:element name="combinationEvent" type="typeGroupCombinationEvent" />

Element typeGroupCombinationEvent / group

Namespace	DR-GW
Diagram	<pre> graph LR group((group)) --> typeSub[typeSubscriberAddress] typeSub --> ssi((ssi)) typeSub --> tsi((tsi)) </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre> <group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group> </pre>
Source	<xss:element name="group" type="typeSubscriberAddress" />

Element typeGroupCombinationEvent / baseGroup

Namespace	DR-GW
Diagram	<pre> graph LR baseGroup((baseGroup)) --> typeSub[typeSubscriberAddress] typeSub --> ssi((ssi)) typeSub --> tsi((tsi)) </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre> <baseGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </baseGroup> </pre>
Source	<xss:element name="baseGroup" type="typeSubscriberAddress" />

Element typeGroupCombinationEvent / constitGroup

Namespace	DR-GW
Diagram	<pre> graph LR constitGroup((constitGroup)) --> typeSub[typeSubscriberAddress] typeSub --> ssi((ssi)) typeSub --> tsi((tsi)) </pre>

Type	typeSubscriberAddress
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: 7</p>
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><constitGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </constitGroup></pre>
Source	<code><xss:element name="constitGroup" type="typeSubscriberAddress" minOccurs="0" maxOccurs="7" /></code>

Element interfaceGroup / addRadioMemberEvent

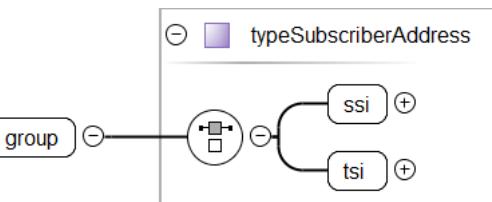
Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeGroupAddRadioMemberEvent typeGroupAddRadioMemberEvent { <> requestId <> result <> radio <> group } </pre>
Type	typeGroupAddRadioMemberEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupAddRadioMemberEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , group
Children	group, radio, requestId, result
Instance	<pre><addRadioMemberEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <group>{1,1}</group> </addRadioMemberEvent></pre>
Source	<code><xss:element name="addRadioMemberEvent" type="typeGroupAddRadioMemberEvent" /></code>

Element typeGroupAddRadioMemberEvent / radio

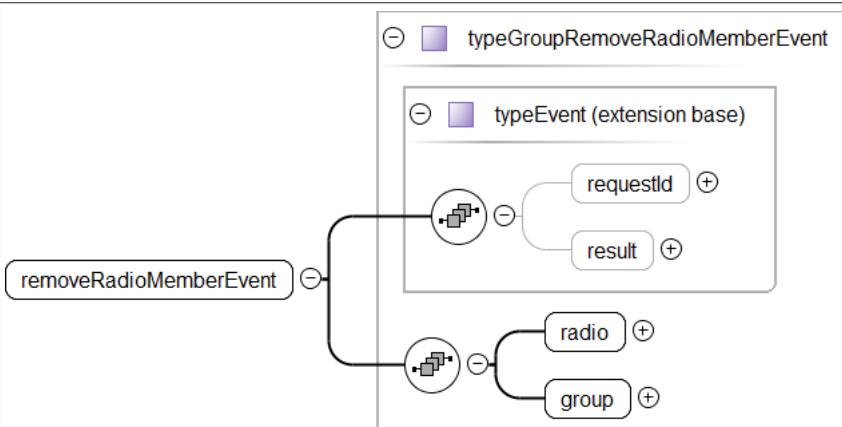
Namespace	DR-GW
Diagram	<pre> typeEvent < -- typeSubscriberAddress typeSubscriberAddress { <> ssi <> tsi } </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi

Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xss:element name="radio" type="typeSubscriberAddress"/>

Element typeGroupAddRadioMemberEvent / group

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xss:element name="group" type="typeSubscriberAddress"/>

Element interfaceGroup / removeRadioMemberEvent

Namespace	DR-GW
Diagram	
Type	typeGroupRemoveRadioMemberEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupRemoveRadioMemberEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , group
Children	group, radio, requestId, result
Instance	<removeRadioMemberEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <group>{1,1}</group> </removeRadioMemberEvent>
Source	<xss:element name="removeRadioMemberEvent" type="typeGroupRemoveRadioMemberEvent"/>

Element typeGroupRemoveRadioMemberEvent / radio

Namespace	DR-GW
Diagram	<pre> classDiagram class radio class typeSubscriberAddress { <<radio>> <<ssi>> <<tsi>> } radio --> typeSubscriberAddress typeSubscriberAddress --> ssi typeSubscriberAddress --> tsi </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xss:element name="radio" type="typeSubscriberAddress"/>

Element typeGroupRemoveRadioMemberEvent / group

Namespace	DR-GW
Diagram	<pre> classDiagram class group class typeSubscriberAddress { <<group>> <<ssi>> <<tsi>> } group --> typeSubscriberAddress typeSubscriberAddress --> ssi typeSubscriberAddress --> tsi </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xss:element name="group" type="typeSubscriberAddress"/>

Element interfaceGroup / addCombinationEvent

Namespace	DR-GW
Diagram	<pre> classDiagram class addCombinationEvent class typeGroupAddCombinationEvent { <<addCombinationEvent>> <<typeEvent (extension base)>> <<requestId>> <<result>> <<group>> <<baseGroup>> } addCombinationEvent --> typeGroupAddCombinationEvent typeGroupAddCombinationEvent --> requestId typeGroupAddCombinationEvent --> result typeGroupAddCombinationEvent --> group typeGroupAddCombinationEvent --> baseGroup </pre>

Type	typeGroupAddCombinationEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupAddCombinationEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , baseGroup
Children	baseGroup, group, requestId, result
Instance	<pre><addCombinationEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> <baseGroup>{1,1}</baseGroup> </addCombinationEvent></pre>
Source	<code><xss:element name="addCombinationEvent" type="typeGroupAddCombinationEvent" /></code>

Element typeGroupAddCombinationEvent / group

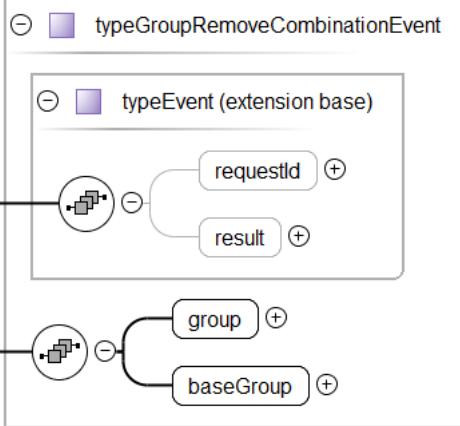
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group></pre>
Source	<code><xss:element name="group" type="typeSubscriberAddress" /></code>

Element typeGroupAddCombinationEvent / baseGroup

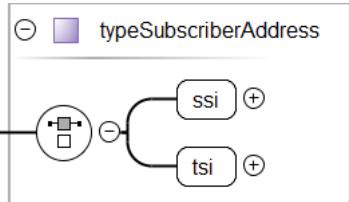
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><baseGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </baseGroup></pre>
Source	<code><xss:element name="baseGroup" type="typeSubscriberAddress" /></code>

Element interfaceGroup / removeCombinationEvent

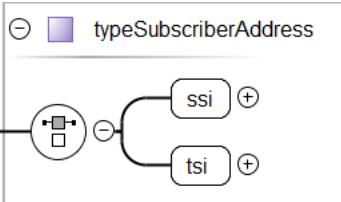
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeGroupRemoveCombinationEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupRemoveCombinationEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , baseGroup
Children	baseGroup, group, requestId, result
Instance	<pre><removeCombinationEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> <baseGroup>{1,1}</baseGroup> </removeCombinationEvent></pre>
Source	<code><xss:element name="removeCombinationEvent" type="typeGroupRemoveCombinationEvent" /></code>

Element typeGroupRemoveCombinationEvent / group

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group></pre>
Source	<code><xss:element name="group" type="typeSubscriberAddress" /></code>

Element typeGroupRemoveCombinationEvent / baseGroup

Namespace	DR-GW
Diagram	

Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<baseGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </baseGroup>
Source	<xs:element name="baseGroup" type="typeSubscriberAddress" />

Element interfaceGroup / subscribeDataEvent

Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeGroupSubscribeDataEvent typeEvent { <<extension base>> requestId result } typeGroupSubscribeDataEvent { <<group>> } typeGroupSubscribeDataEvent "1..1" o--> subscribeDataEvent typeGroupSubscribeDataEvent --> group </pre>
Type	typeGroupSubscribeDataEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupSubscribeDataEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group
Children	group, requestId, result
Instance	<subscribeDataEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> </subscribeDataEvent>
Source	<xs:element name="subscribeDataEvent" type="typeGroupSubscribeDataEvent" />

Element typeGroupSubscribeDataEvent / group

Namespace	DR-GW
Diagram	<pre> classDiagram typeGroupDataSubscription { <<group>> addr useSDS useStatus } group --> typeGroupDataSubscription addr --> typeGroupDataSubscription useSDS --> typeGroupDataSubscription useStatus --> typeGroupDataSubscription </pre>
Type	typeGroupDataSubscription
Properties	content: complex
Model	addr , useSDS , useStatus
Children	addr, useSDS, useStatus
Instance	<group xmlns="DR-GW"> <addr>{1,1}</addr>

	<pre><useSDS>{1,1}</useSDS> <useStatus>{1,1}</useStatus> </group></pre>
Source	<code><xss:element name="group" type="typeGroupDataSubscription" /></code>

Element interfaceGroup / event

Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeGroupEvent event --> typeGroupEvent typeGroupEvent "0..1" requestId typeGroupEvent "0..1" result typeGroupEvent "1..1" group typeGroupEvent "1..1" delete </pre>
Type	typeGroupEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , delete
Children	delete, group, requestId, result
Instance	<pre> <event xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> <delete>{1,1}</delete> </event> </pre>
Source	<code><xss:element name="event" type="typeGroupEvent" /></code>

Element typeGroupEvent / group

Namespace	DR-GW
Diagram	<pre> typeGroup "0..1" group typeGroup "0..1" addr typeGroup "0..1" alias typeGroup "0..1" orgblockId </pre>
Type	typeGroup
Properties	content: complex
Model	addr , alias , orgblockId
Children	addr, alias, orgblockId
Instance	<pre> <group xmlns="DR-GW"> <addr>{1,1}</addr> <alias>{1,1}</alias> <orgblockId>{1,1}</orgblockId> </group> </pre>
Source	<code><xss:element name="group" type="typeGroup" /></code>

Element typeGroupEvent / delete

Namespace	DR-GW
Diagram	<p>Diagram illustrating the 'delete' element:</p> <pre> graph LR delete([delete]) --> xsBoolean(xs:boolean) </pre> <p>Built-in primitive type. It defines the boolean values true and false.</p>
Type	xs:boolean
Properties	content: simple
Source	<code><xs:element name="delete" type="xs:boolean"/></code>

Element drgw / radio

Namespace	DR-GW
Diagram	<p>Diagram illustrating the 'radio' element:</p> <pre> graph LR radio[radio Type interfaceRadio] --> interfaceRadio(interfaceRadio) </pre> <p>DR-GW-Radio. Use to get information about certain radios or to read out the complete radio list the DF-Client has...</p>
Type	interfaceRadio

Properties	content: complex
Model	get getList getGroups track changeOpta enable disable response getEvent getListEvent getGroupsEvent trackSubscriptionEvent trackEvent groupsEvent changeOptaEvent enableDisableEvent event
Children	changeOpta, changeOptaEvent, disable, enable, enableDisableEvent, event, get, getEvent, getGroups, getGroupsEvent, getList, getListEvent, groupsEvent, response, track, trackEvent, trackSubscriptionEvent
Instance	<pre><radio xmlns="DR-GW"> <get>{1,1}</get> <getList>{1,1}</getList> <getGroups>{1,1}</getGroups> <track>{1,1}</track> <changeOpta>{1,1}</changeOpta> <enable>{1,1}</enable> <disable>{1,1}</disable> <response>{1,1}</response> <getEvent>{1,1}</getEvent> <getListEvent>{1,1}</getListEvent> <getGroupsEvent>{1,1}</getGroupsEvent> <trackSubscriptionEvent>{1,1}</trackSubscriptionEvent> <trackEvent>{1,1}</trackEvent> <groupsEvent>{1,1}</groupsEvent> <changeOptaEvent>{1,1}</changeOptaEvent> <enableDisableEvent>{1,1}</enableDisableEvent> <event>{1,1}</event> </radio></pre>
Source	<code><xss:element name="radio" type="interfaceRadio"/></code>

Element interfaceRadio / get

Namespace	DR-GW
Diagram	<pre> classDiagram typeRadioGet < -- typeRequest typeRadioGet "1..1" -- "1..1" requestId typeRadioGet "1..1" -- "1..1" radio </pre>
Type	typeRadioGet
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeRadioGet
Properties	content: complex
Model	requestId , radio
Children	radio, requestId
Instance	<pre><get xmlns="DR-GW"> <requestId>{1,1}</requestId> <radio>{1,1}</radio> </get></pre>
Source	<code><xss:element name="get" type="typeRadioGet"/></code>

Element typeRadioGet / radio

Namespace	DR-GW
Diagram	<pre> classDiagram typeSubscriberAddress "1..1" -- "1..1" radio typeSubscriberAddress "1..1" -- "1..1" ssi typeSubscriberAddress "1..1" -- "1..1" tsi </pre>

Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xss:element name="radio" type="typeSubscriberAddress"/>

Element interfaceRadio / getList

Namespace	DR-GW
Diagram	<pre> classDiagram typeRadioGetList < -- typeRequest typeRadioGetList "1..1" --> getList typeRadioGetList "1..1" --> orgblockId typeRadioGetList "1..1" --> requestID </pre>
Type	typeRadioGetList
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeRadioGetList
Properties	content: complex
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId
Instance	<getList xmlns="DR-GW"> <requestId>{1,1}</requestId> <orgblockId>{0,1}</orgblockId> </getList>
Source	<xss:element name="getList" type="typeRadioGetList"/>

Element typeRadioGetList / orgblockID

Namespace	DR-GW				
Diagram	<pre> classDiagram typeOrganisationBlockId "1..1" --> orgblockID typeOrganisationBlockId "1..1" --> orgblockIDSimple </pre>				
Type	typeOrganisationBlockId				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	orgblockID orgblockIDSimple				
Children	orgblockID, orgblockIDSimple				
Instance	<orgblockID xmlns="DR-GW"> <orgblockID>{1,1}</orgblockID> <orgblockIDSimple>{1,1}</orgblockIDSimple> </orgblockID>				
Source	<xss:element name="orgblockID" type="typeOrganisationBlockId" minOccurs="0"/>				

Element interfaceRadio / getGroups

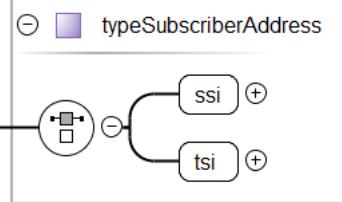
Namespace	DR-GW
Diagram	<pre> classDiagram typeRadioGetGroups "1..1" -- "*" getGroups typeRadioGetGroups "*" -- "*" radio typeRadioGetGroups < -- typeRequest </pre>
Type	typeRadioGetGroups
Type hierarchy	<ul style="list-style-type: none"> • typeRequest <ul style="list-style-type: none"> • typeRadioGetGroups
Properties	content: complex
Model	requestId , radio
Children	radio, requestId
Instance	<pre> <getGroups xmlns="DR-GW"> <requestId>{1,1}</requestId> <radio>{1,1}</radio> </getGroups> </pre>
Source	<pre> <xss:element name="getGroups" type="typeRadioGetGroups"/> </pre>

Element typeRadioGetGroups / radio

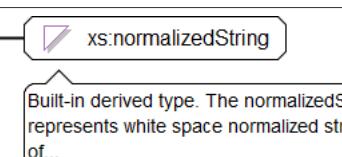
Namespace	DR-GW
Diagram	<pre> classDiagram typeRadio "*" -- "1..1" issi typeRadio "*" -- "1..1" alias typeRadio "*" -- "1..1" orgblockId typeRadio "*" -- "0..1" opta </pre>
Type	typeRadio
Properties	content: complex
Model	issi , alias , orgblockId , opta{0,1}
Children	alias, issi, opta, orgblockId
Instance	<pre> <radio xmlns="DR-GW"> <issi>{1,1}</issi> <alias>{1,1}</alias> <orgblockId>{1,1}</orgblockId> <opta>{0,1}</opta> </radio> </pre>
Source	<pre> <xss:element name="radio" type="typeRadio"/> </pre>

Element typeRadio / issi

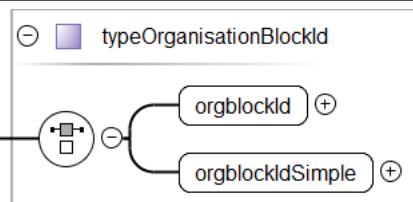
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><issi xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </issi></pre>
Source	<code><xss:element name="issi" type="typeSubscriberAddress" /></code>

Element typeRadio / alias

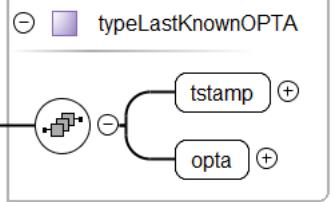
Namespace	DR-GW
Diagram	 A callout box points to the xs:normalizedString association with the text: "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of..."
Type	xs:normalizedString
Properties	content: simple
Source	<code><xss:element name="alias" type="xs:normalizedString" /></code>

Element typeRadio / orgblockId

Namespace	DR-GW
Diagram	
Type	typeOrganisationBlockId
Properties	content: complex
Model	orgblockId orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<pre><orgblockId xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <orgblockIdSimple>{1,1}</orgblockIdSimple> </orgblockId></pre>
Source	<code><xss:element name="orgblockId" type="typeOrganisationBlockId" /></code>

Element typeRadio / opta

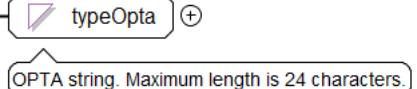
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeLastKnownOPTA
Properties	<p>content: complex</p> <p>minOccurs: 0</p>
Model	tstamp , opta
Children	opta, tstamp
Instance	<pre><opta xmlns="DR-GW"> <tstamp>{1,1}</tstamp> <opta>{1,1}</opta> </opta></pre>
Source	<code><xss:element name="opta" type="typeLastKnownOPTA" minOccurs="0" /></code>

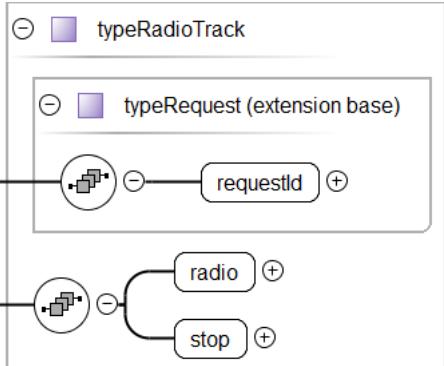
Element typeLastKnownOPTA / tstamp

Namespace	DR-GW
Diagram	 Built-in primitive type. The dateTime datatype represents a specific instant of time.
Type	xs:dateTime
Properties	content: simple
Source	<code><xss:element name="tstamp" type="xs:dateTime" /></code>

Element typeLastKnownOPTA / opta

Namespace	DR-GW
Diagram	 OPTA string. Maximum length is 24 characters.
Type	typeOpta
Properties	content: simple
Facets	maxLength 24
Source	<code><xss:element name="opta" type="typeOpta" /></code>

Element interfaceRadio / track

Namespace	DR-GW
Diagram	

Type	typeRadioTrack
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeRadioTrack
Properties	content: complex
Model	requestId , radio , stop
Children	radio, requestId, stop
Instance	<pre><track xmlns="DR-GW"> <requestId>{1,1}</requestId> <radio>{1,1}</radio> <stop>{1,1}</stop> </track></pre>
Source	<code><xss:element name="track" type="typeRadioTrack" /></code>

Element typeRadioTrack / radio

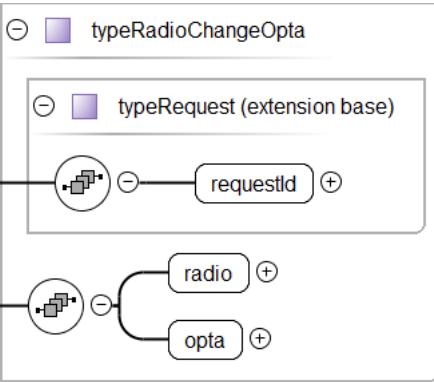
Namespace	DR-GW
Diagram	<pre> classDiagram class typeRadioTrack { radio requestId stop } radio --> requestId radio --> stop requestId +- stop -+ </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio></pre>
Source	<code><xss:element name="radio" type="typeSubscriberAddress" /></code>

Element typeRadioTrack / stop

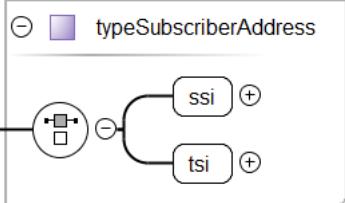
Namespace	DR-GW
Diagram	<pre> classDiagram class typeRadioTrack { stop } stop -+ </pre> <p>Built-in primitive type. It defines the boolean values true and false.</p>
Type	xs:boolean
Properties	content: simple
Source	<code><xss:element name="stop" type="xs:boolean" /></code>

Element interfaceRadio / changeOpta

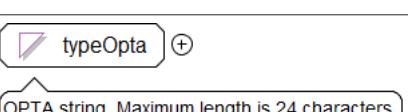
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeRadioChangeOpta
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeRadioChangeOpta
Properties	content: complex
Model	requestId , radio , opta
Children	opta, radio, requestId
Instance	<pre><changeOpta xmlns="DR-GW"> <requestIds>{1,1}</requestIds> <radio>{1,1}</radio> <opta>{1,1}</opta> </changeOpta></pre>
Source	<code><xss:element name="changeOpta" type="typeRadioChangeOpta"/></code>

Element typeRadioChangeOpta / radio

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio></pre>
Source	<code><xss:element name="radio" type="typeSubscriberAddress"/></code>

Element typeRadioChangeOpta / opta

Namespace	DR-GW
Diagram	
Type	typeOpta
Properties	content: simple
Facets	maxLength 24

Source	<code><xss:element name="opta" type="typeOpta"/></code>
--------	---

Element interfaceRadio / enable

Namespace	DR-GW
Diagram	<pre> classDiagram typeRequest < -- typeRadioEnable typeRadioEnable { <<enable>> <<radio>> <<reason>> <<enable>> } note over typeRadioEnable: This method is used to Enable the radio terminal over the air. </pre>
Type	typeRadioEnable
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeRadioEnable
Properties	content: complex
Model	requestId , radio , radio , reason{0,1} , enable
Children	enable, radio, reason, requestId
Instance	<pre> <enable xmlns="DR-GW"> <requestId>{1,1}</requestId> <radio>{1,1}</radio> <radio>{1,1}</radio> <reason>{0,1}</reason> <enable>{1,1}</enable> </enable> </pre>
Source	<code><xss:element name="enable" type="typeRadioEnable"/></code>

Element typeRadioEnable / radio

Namespace	DR-GW
Diagram	<pre> typeRadio < -- typeSubscriberAddress typeSubscriberAddress { <<radio>> <<ssi>> <<tsi>> } </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre> <radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio> </pre>
Source	<code><xss:element name="radio" type="typeSubscriberAddress"/></code>

Element typeRadioEnable / reason

Namespace	DR-GW				
Diagram	<p>Diagram illustrating the element 'reason' with a minOccurs=0 constraint. A callout box provides a detailed explanation of the xs:unsignedByte datatype.</p>				
Type	xs:unsignedByte				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<xss:element name="reason" type="xs:unsignedByte" minOccurs="0" />				

Element typeRadioEnable / enable

Namespace	DR-GW		
Diagram	<p>Diagram illustrating the element 'enable' with a minOccurs=0 constraint. A callout box provides a detailed explanation of the xs:boolean datatype.</p>		
Type	xs:boolean		
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple
content:	simple		
Source	<xss:element name="enable" type="xs:boolean" />		

Element interfaceRadio / disable

Namespace	DR-GW
Diagram	<p>Diagram illustrating the typeRadioDisable element, which extends typeRequest. It shows the structure of the disable method, including requestID, radio, radio, reason, and enable elements. A callout box provides a detailed explanation of the disable method's purpose.</p>
Type	typeRadioDisable
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeRadioDisable
Properties	content: complex
Model	requestId , radio , radio , reason{0,1} , enable
Children	enable, radio, reason, requestId

Instance	<pre><disable xmlns="DR-GW"> <requestId>{1,1}</requestId> <radio>{1,1}</radio> <radio>{1,1}</radio> <reason>{0,1}</reason> <enable>{1,1}</enable> </disable></pre>
Source	<pre><xss:element name="disable" type="typeRadioDisable" /></pre>

Element typeRadioDisable / radio

Namespace	DR-GW
Diagram	<pre> classDiagram class typeRadioDisable { <<radio>> --> radio radio --> ssi radio --> tsi } </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio></pre>
Source	<pre><xss:element name="radio" type="typeSubscriberAddress" /></pre>

Element typeRadioDisable / reason

Namespace	DR-GW
Diagram	<pre> classDiagram class typeRadioDisable { <<reason>> --> xs:unsignedByte } </pre> <p>Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...</p>
Type	xs:unsignedByte
Properties	content: simple minOccurs: 0
Source	<pre><xss:element name="reason" type="xs:unsignedByte" minOccurs="0" /></pre>

Element typeRadioDisable / enable

Namespace	DR-GW
Diagram	<pre> classDiagram class typeRadioDisable { <<enable>> --> xs:boolean } </pre> <p>Built-in primitive type. It defines the boolean values true and false.</p>
Type	xs:boolean
Properties	content: simple
Source	<pre><xss:element name="enable" type="xs:boolean" /></pre>

Element interfaceRadio / response

Namespace	DR-GW
-----------	-------

Diagram	
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<pre><response xmlns="DR-GW"> <requestIds>{1,1}</requestIds> <result>{1,1}</result> </response></pre>
Source	<code><xss:element name="response" type="typeResponse" /></code>

Element interfaceRadio / getEvent

Namespace	DR-GW
Diagram	
Type	typeRadioGetEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioGetEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio
Children	radio, requestId, result
Instance	<pre><getEvent xmlns="DR-GW"> <requestIds>{0,1}</requestIds> <result>{0,1}</result> <radio>{1,1}</radio> </getEvent></pre>
Source	<code><xss:element name="getEvent" type="typeRadioGetEvent" /></code>

Element typeRadioGetEvent / radio

Namespace	DR-GW
-----------	-------

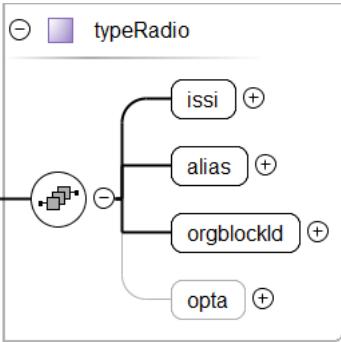
Diagram	
Type	typeRadio
Properties	content: complex
Model	issi , alias , orgblockId , opta{0,1}
Children	alias, issi, opta, orgblockId
Instance	<pre><radio xmlns="DR-GW"> <issi>{1,1}</issi> <alias>{1,1}</alias> <orgblockId>{1,1}</orgblockId> <opta>{0,1}</opta> </radio></pre>
Source	<code><xss:element name="radio" type="typeRadio"/></code>

Element interfaceRadio / getListEvent

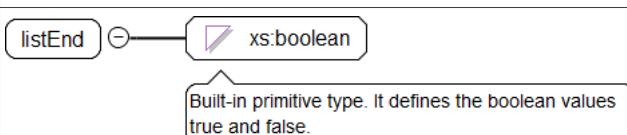
Namespace	DR-GW
Diagram	
Type	typeRadioGetListEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioGetListEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio* , listEnd
Children	listEnd, radio, requestId, result
Instance	<pre><getListEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{0..unbounded}</radio> <listEnd>{1,1}</listEnd> </getListEvent></pre>
Source	<code><xss:element name="getListEvent" type="typeRadioGetListEvent"/></code>

Element typeRadioGetListEvent / radio

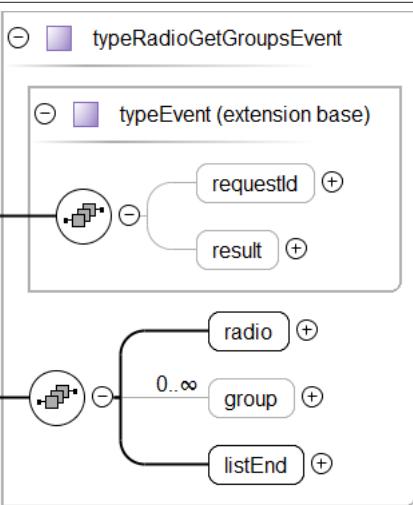
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeRadio
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Model	issi , alias , orgblockId , opta{0,1}
Children	alias, issi, opta, orgblockId
Instance	<pre><radio xmlns="DR-GW"> <issi>{1,1}</issi> <alias>{1,1}</alias> <orgblockId>{1,1}</orgblockId> <opta>{0,1}</opta> </radio></pre>
Source	<code><x:element name="radio" type="typeRadio" minOccurs="0" maxOccurs="unbounded"/></code>

Element typeRadioGetListEvent / listEnd

Namespace	DR-GW
Diagram	 Built-in primitive type. It defines the boolean values true and false.
Type	xs:boolean
Properties	content: simple
Source	<code><x:element name="listEnd" type="xs:boolean"/></code>

Element interfaceRadio / getGroupsEvent

Namespace	DR-GW
Diagram	
Type	typeRadioGetGroupsEvent

Type hierarchy	<ul style="list-style-type: none"> • typeEvent <ul style="list-style-type: none"> • typeRadioGetGroupsEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , group* , listEnd
Children	group, listEnd, radio, requestId, result
Instance	<pre><getGroupsEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <group>{0,unbounded}</group> <listEnd>{1,1}</listEnd> </getGroupsEvent></pre>
Source	<code><xss:element name="getGroupsEvent" type="typeRadioGetGroupsEvent" /></code>

Element typeRadioGetGroupsEvent / radio

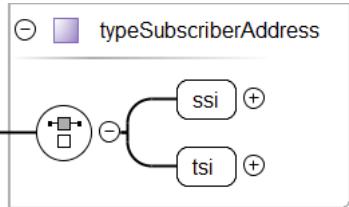
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio></pre>
Source	<code><xss:element name="radio" type="typeSubscriberAddress" /></code>

Element typeRadioGetGroupsEvent / group

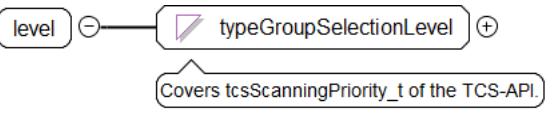
Namespace	DR-GW						
Diagram							
Type	typeRadioGroupSelection						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
content:	complex						
minOccurs:	0						
maxOccurs:	unbounded						
Model	group , level						
Children	group, level						
Instance	<pre><group xmlns="DR-GW"> <group>{1,1}</group> <level>{1,1}</level> </group></pre>						
Source	<code><xss:element name="group" type="typeRadioGroupSelection" minOccurs="0" maxOccurs="unbounded" /></code>						

Element typeRadioGroupSelection / group

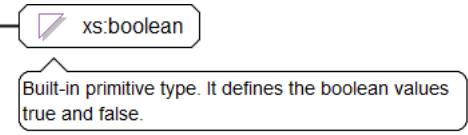
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group></pre>
Source	<code><xss:element name="group" type="typeSubscriberAddress"/></code>

Element typeRadioGroupSelection / level

Namespace	DR-GW
Diagram	
Type	typeGroupSelectionLevel
Properties	content: simple
Facets	enumeration notScanned enumeration low enumeration normal enumeration selected enumeration high enumeration background
Source	<code><xss:element name="level" type="typeGroupSelectionLevel"/></code>

Element typeRadioGetGroupsEvent / listEnd

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<code><xss:element name="listEnd" type="xs:boolean"/></code>

Element interfaceRadio / trackSubscriptionEvent

Namespace	DR-GW
-----------	-------

Diagram	
Type	typeRadioTrackSubscriptionEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioTrackSubscriptionEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , stop
Children	radio, requestId, result, stop
Instance	<pre><trackSubscriptionEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <stop>{1,1}</stop> </trackSubscriptionEvent></pre>
Source	<code><xss:element name="trackSubscriptionEvent" type="typeRadioTrackSubscriptionEvent" /></code>

Element typeRadioTrackSubscriptionEvent / radio

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio></pre>
Source	<code><xss:element name="radio" type="typeSubscriberAddress" /></code>

Element typeRadioTrackSubscriptionEvent / stop

Namespace	DR-GW
Diagram	
Type	xs:boolean

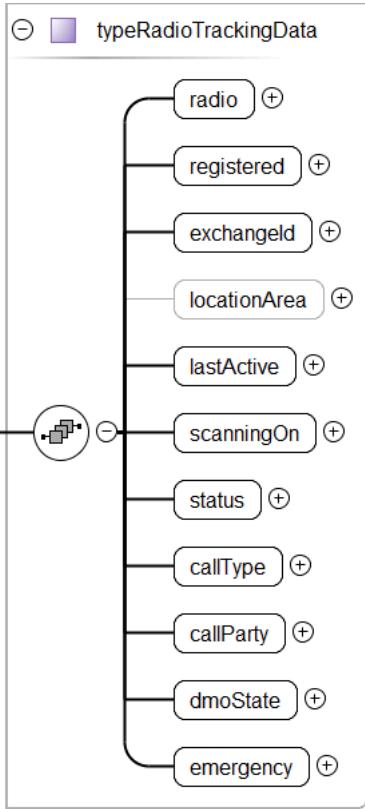
Properties	content: simple
Source	<xs:element name="stop" type="xs:boolean"/>

Element interfaceRadio / trackEvent

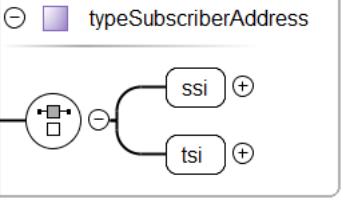
Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeRadioTrackEvent typeRadioTrackEvent { -requestId -result +trackingData } </pre>
Type	typeRadioTrackEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioTrackEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , trackingData
Children	requestId, result, trackingData
Instance	<pre> <trackEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <trackingData>{1,1}</trackingData> </trackEvent> </pre>
Source	<xs:element name="trackEvent" type="typeRadioTrackEvent"/>

Element typeRadioTrackEvent / trackingData

Namespace	DR-GW
-----------	-------

Diagram	
Type	typeRadioTrackingData
Properties	content: complex
Model	radio , registered , exchangeId , locationArea{0,1} , lastActive , scanningOn , status , callType , callParty , dmoState , emergency
Children	callParty, callType, dmoState, emergency, exchangeId, lastActive, locationArea, radio, registered, scanningOn, status
Instance	<pre><trackingData xmlns="DR-GW"> <radio>{1,1}</radio> <registered>{1,1}</registered> <exchangeId>{1,1}</exchangeId> <locationArea>{0,1}</locationArea> <lastActive>{1,1}</lastActive> <scanningOn>{1,1}</scanningOn> <status>{1,1}</status> <callType>{1,1}</callType> <callParty>{1,1}</callParty> <dmoState>{1,1}</dmoState> <emergency>{1,1}</emergency> </trackingData></pre>
Source	<code><xs:element name="trackingData" type="typeRadioTrackingData"/></code>

Element typeRadioTrackingData / radio

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi

Instance	<pre><radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio></pre>
Source	<pre><xs:element name="radio" type="typeSubscriberAddress" /></pre>

Element typeRadioTrackingData / registered

Namespace	DR-GW
Diagram	<pre> graph LR registered[registered] --> xsBoolean[xs:boolean] </pre> <p>Built-in primitive type. It defines the boolean values true and false.</p>
Type	xs:boolean
Properties	content: simple
Source	<pre><xs:element name="registered" type="xs:boolean" /></pre>

Element typeRadioTrackingData / exchangeId

Namespace	DR-GW
Diagram	<pre> graph LR exchangeId[exchangeId] --> xsUnsignedLong(xs:unsignedLong) </pre> <p>Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...</p>
Type	xs:unsignedLong
Properties	content: simple
Source	<pre><xs:element name="exchangeId" type="xs:unsignedLong" /></pre>

Element typeRadioTrackingData / locationArea

Namespace	DR-GW
Diagram	<pre> graph LR locationArea[locationArea] --> xsUnsignedShort(xs:unsignedShort) </pre> <p>Built-in derived type. The unsignedShort datatype is derived from unsignedInt by setting the value of maxInclusive to...</p>
Type	xs:unsignedShort
Properties	content: simple minOccurs: 0
Source	<pre><xs:element name="locationArea" type="xs:unsignedShort" minOccurs="0" /></pre>

Element typeRadioTrackingData / lastActive

Namespace	DR-GW
Diagram	<pre> graph LR lastActive[lastActive] --> xsDateTime(xs:dateTime) </pre> <p>Built-in primitive type. The dateTime datatype represents a specific instant of time.</p>
Type	xs:dateTime
Properties	content: simple
Source	<pre><xs:element name="lastActive" type="xs:dateTime" /></pre>

Element typeRadioTrackingData / scanningOn

Namespace	DR-GW
Diagram	<p>The diagram shows a rounded rectangle labeled "scanningOn" connected by a line to a purple icon representing "xs:boolean". A callout box below the connection states: "Built-in primitive type. It defines the boolean values true and false."</p>
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="scanningOn" type="xs:boolean" />

Element typeRadioTrackingData / status

Namespace	DR-GW
Diagram	<p>The diagram shows a rounded rectangle labeled "status" connected by a line to a purple icon representing "typeStatusIndicator". Inside "typeStatusIndicator", there is a "value" node and a "time" node, both with plus signs indicating they are children.</p>
Type	typeStatusIndicator
Properties	content: complex
Model	value , time
Children	time, value
Instance	<status xmlns="DR-GW"> <value>{1,1}</value> <time>{1,1}</time> </status>
Source	<xs:element name="status" type="typeStatusIndicator" />

Element typeStatusIndicator / value

Namespace	DR-GW
Diagram	<p>The diagram shows a rounded rectangle labeled "value" connected by a line to a purple icon representing "xs:unsignedLong". A callout box below the connection states: "Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of..."</p>
Type	xs:unsignedLong
Properties	content: simple
Source	<xs:element name="value" type="xs:unsignedLong" />

Element typeStatusIndicator / time

Namespace	DR-GW
Diagram	<p>The diagram shows a rounded rectangle labeled "time" connected by a line to a purple icon representing "xs:dateTime". A callout box below the connection states: "Built-in primitive type. The dateTime datatype represents a specific instant of time."</p>
Type	xs:dateTime
Properties	content: simple
Source	<xs:element name="time" type="xs:dateTime" />

Element typeRadioTrackingData / callType

Namespace	DR-GW						
Diagram	<pre> graph LR callType([callType]) --> typeCallType[/typeCallType/] typeCallType --> callout[Call type attribute. Choices are Point2Point, Point2MultiPoint or Broadcast.] </pre>						
Type	typeCallType						
Properties	content: simple						
Facets	<table border="1"> <tr> <td>enumeration</td> <td>p2p</td> </tr> <tr> <td>enumeration</td> <td>p2mp</td> </tr> <tr> <td>enumeration</td> <td>bcast</td> </tr> </table>	enumeration	p2p	enumeration	p2mp	enumeration	bcast
enumeration	p2p						
enumeration	p2mp						
enumeration	bcast						
Source	<code><xss:element name="callType" type="typeCallType" /></code>						

Element typeRadioTrackingData / callParty

Namespace	DR-GW
Diagram	<pre> graph LR callParty([callParty]) --> typeSubscriberAddress[/typeSubscriberAddress/] typeSubscriberAddress --> ssi[ssi] typeSubscriberAddress --> tsi[tsi] </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<code><callParty xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </callParty></code>
Source	<code><xss:element name="callParty" type="typeSubscriberAddress" /></code>

Element typeRadioTrackingData / dmoState

Namespace	DR-GW
Diagram	<pre> graph LR dmoState([dmoState]) --> xsBoolean[/xs:boolean/] xsBoolean --> callout[Built-in primitive type. It defines the boolean values true and false.] </pre>
Type	xs:boolean
Properties	content: simple
Source	<code><xss:element name="dmoState" type="xs:boolean" /></code>

Element typeRadioTrackingData / emergency

Namespace	DR-GW
Diagram	<pre> graph LR emergency([emergency]) --> xsBoolean[/xs:boolean/] xsBoolean --> callout[Built-in primitive type. It defines the boolean values true and false.] </pre>

Type	xs:boolean
Properties	content: simple
Source	<xss:element name="emergency" type="xs:boolean"/>

Element interfaceRadio / groupsEvent

Namespace	DR-GW
Diagram	<pre> classDiagram typeRadioGroupsEvent < -- typeEvent typeEvent < -- groupsEvent groupsEvent < -- requestId groupsEvent < -- result groupsEvent --> radio radio --> group radio --> deletedGroup </pre>
Type	typeRadioGroupsEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioGroupsEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , (group+ deletedGroup)
Children	deletedGroup, group, radio, requestId, result
Instance	<pre> <groupsEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <group>{1,unbounded}</group> <deletedGroup>{1,1}</deletedGroup> </groupsEvent> </pre>
Source	<xss:element name="groupsEvent" type="typeRadioGroupsEvent"/>

Element typeRadioGroupsEvent / radio

Namespace	DR-GW
Diagram	<pre> typeRadioGroupsEvent < -- typeSubscriberAddress typeSubscriberAddress < -- radio radio < -- ssi radio < -- tsi </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre> <radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio> </pre>
Source	<xss:element name="radio" type="typeSubscriberAddress"/>

Element typeRadioGroupsEvent / group

Namespace	DR-GW				
Diagram	<pre> classDiagram typeRadioGroupSelection < -- group typeRadioGroupSelection < -- level typeRadioGroupSelection < -- group </pre>				
Type	typeRadioGroupSelection				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	maxOccurs:	unbounded
content:	complex				
maxOccurs:	unbounded				
Model	group , level				
Children	group, level				
Instance	<group xmlns="DR-GW"> <group>{1,1}</group> <level>{1,1}</level> </group>				
Source	<xss:element name="group" type="typeRadioGroupSelection" maxOccurs="unbounded"/>				

Element typeRadioGroupsEvent / deletedGroup

Namespace	DR-GW		
Diagram	<pre> classDiagram typeSubscriberAddress < -- deletedGroup typeSubscriberAddress < -- ssi typeSubscriberAddress < -- tsi </pre>		
Type	typeSubscriberAddress		
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> </table>	content:	complex
content:	complex		
Model	ssi tsi		
Children	ssi, tsi		
Instance	<deletedGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </deletedGroup>		
Source	<xss:element name="deletedGroup" type="typeSubscriberAddress"/>		

Element interfaceRadio / changeOptaEvent

Namespace	DR-GW
Diagram	<pre> classDiagram typeRadioChangeOptaEvent < -- changeOptaEvent typeRadioChangeOptaEvent < -- radio typeRadioChangeOptaEvent < -- opta </pre>

Type	typeRadioChangeOptaEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioChangeOptaEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , opta
Children	opta, radio, requestId, result
Instance	<pre><changeOptaEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <opta>{1,1}</opta> </changeOptaEvent></pre>
Source	<code><xss:element name="changeOptaEvent" type="typeRadioChangeOptaEvent" /></code>

Element typeRadioChangeOptaEvent / radio

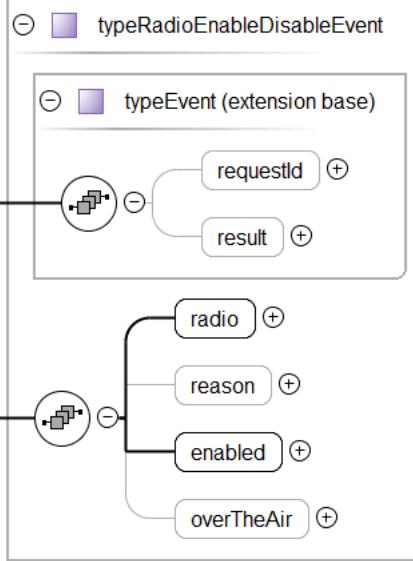
Namespace	DR-GW
Diagram	<pre> classDiagram class typeSubscriberAddress { radio ssi tsi } radio --> typeSubscriberAddress ssi +--> typeSubscriberAddress tsi +--> typeSubscriberAddress </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio></pre>
Source	<code><xss:element name="radio" type="typeSubscriberAddress" /></code>

Element typeRadioChangeOptaEvent / opta

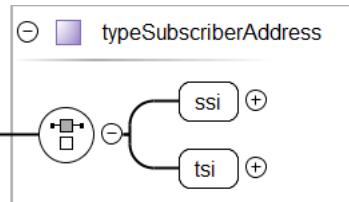
Namespace	DR-GW
Diagram	<pre> classDiagram class typeOpta { opta } opta --> typeOpta </pre> <p>OPTA string. Maximum length is 24 characters.</p>
Type	typeOpta
Properties	content: simple
Facets	maxLength 24
Source	<code><xss:element name="opta" type="typeOpta" /></code>

Element interfaceRadio / enableDisableEvent

Namespace	DR-GW
-----------	-------

Diagram	
Type	typeRadioEnableDisableEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioEnableDisableEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , reason{0,1} , enabled , overTheAir{0,1}
Children	enabled, overTheAir, radio, reason, requestId, result
Instance	<pre><enableDisableEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <reason>{0,1}</reason> <enabled>{1,1}</enabled> <overTheAir>{0,1}</overTheAir> </enableDisableEvent></pre>
Source	<code><xs:element name="enableDisableEvent" type="typeRadioEnableDisableEvent" /></code>

Element typeRadioEnableDisableEvent / radio

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio></pre>
Source	<code><xs:element name="radio" type="typeSubscriberAddress" /></code>

Element typeRadioEnableDisableEvent / reason

Namespace	DR-GW
-----------	-------

Diagram	<p>xs:unsignedByte</p> <p>Built-in derived type. The unsignedByte datatype is derived from unsignedShort by setting the value of maxInclusive to...</p>
Type	xs:unsignedByte
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xs:element name="reason" type="xs:unsignedByte" minOccurs="0" />

Element typeRadioEnableDisableEvent / enabled

Namespace	DR-GW
Diagram	<p>xs:boolean</p> <p>Built-in primitive type. It defines the boolean values true and false.</p>
Type	xs:boolean
Properties	<p>content: simple</p>
Source	<xs:element name="enabled" type="xs:boolean" />

Element typeRadioEnableDisableEvent / overTheAir

Namespace	DR-GW
Diagram	<p>xs:boolean</p> <p>Built-in primitive type. It defines the boolean values true and false.</p>
Type	xs:boolean
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xs:element name="overTheAir" type="xs:boolean" minOccurs="0" />

Element interfaceRadio / event

Namespace	DR-GW
Diagram	<p>typeRadioEvent</p> <p>typeEvent (extension base)</p> <p>requestId</p> <p>result</p> <p>radio</p> <p>delete</p>
Type	typeRadioEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent

	<ul style="list-style-type: none"> • typeRadioEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , delete
Children	delete, radio, requestId, result
Instance	<pre><event xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <delete>{1,1}</delete> </event></pre>
Source	<code><xss:element name="event" type="typeRadioEvent" /></code>

Element typeRadioEvent / radio

Namespace	DR-GW
Diagram	<pre> classDiagram typeRadioEvent { radio } radio { issi alias orgblockId opta } </pre>
Type	typeRadio
Properties	content: complex
Model	issi , alias , orgblockId , opta{0,1}
Children	alias, issi, opta, orgblockId
Instance	<pre><radio xmlns="DR-GW"> <issi>{1,1}</issi> <alias>{1,1}</alias> <orgblockId>{1,1}</orgblockId> <opta>{0,1}</opta> </radio></pre>
Source	<code><xss:element name="radio" type="typeRadio" /></code>

Element typeRadioEvent / delete

Namespace	DR-GW
Diagram	<p>Built-in primitive type. It defines the boolean values true and false.</p>
Type	xs:boolean
Properties	content: simple
Source	<code><xss:element name="delete" type="xs:boolean" /></code>

Element drgw / app

Namespace	DR-GW
-----------	-------

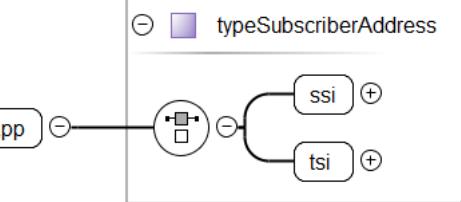
Diagram	<p>DR-GW-Application. Use to get information about certain client application or to read out the complete client...</p>
Type	interfaceApp
Properties	content: complex
Model	get getList response getEvent getListEvent
Children	get, getEvent, getList, getListEvent, response
Instance	<pre><app xmlns="DR-GW"> <get>{1,1}</get> <getList>{1,1}</getList> <response>{1,1}</response> <getEvent>{1,1}</getEvent> <getListEvent>{1,1}</getListEvent> </app></pre>
Source	<code><xss:element name="app" type="interfaceApp" /></code>

Element interfaceApp / get

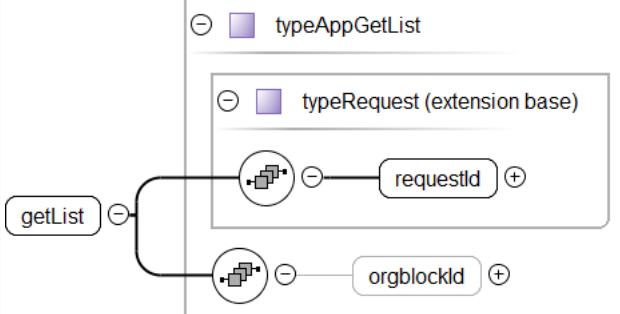
Namespace	DR-GW
Diagram	
Type	typeAppGet
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeAppGet
Properties	content: complex
Model	requestId , app
Children	app, requestId
Instance	<pre><get xmlns="DR-GW"> <requestId>{1,1}</requestId> <app>{1,1}</app> </get></pre>
Source	<code><xss:element name="get" type="typeAppGet" /></code>

Element typeAppGet / app

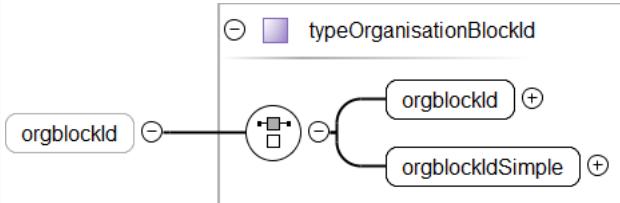
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<app xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </app>
Source	<xss:element name="app" type="typeSubscriberAddress"/>

Element interfaceApp / getList

Namespace	DR-GW
Diagram	
Type	typeAppGetList
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeAppGetList
Properties	content: complex
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId
Instance	<getList xmlns="DR-GW"> <requestId>{1,1}</requestId> <orgblockId>{0,1}</orgblockId> </getList>
Source	<xss:element name="getList" type="typeAppGetList"/>

Element typeAppGetList / orgblockId

Namespace	DR-GW				
Diagram					
Type	typeOrganisationBlockId				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				

Model	orgblockId orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<pre><orgblockId xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <orgblockIdSimple>{1,1}</orgblockIdSimple> </orgblockId></pre>
Source	<code><xs:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0" /></code>

Element interfaceApp / response

Namespace	DR-GW
Diagram	<p>typeResponse</p> <p>response</p> <p>requestId</p> <p>result</p> <p>Response contains result of execution of any method.</p>
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<pre><response xmlns="DR-GW"> <requestId>{1,1}</requestId> <result>{1,1}</result> </response></pre>
Source	<code><xs:element name="response" type="typeResponse" /></code>

Element interfaceApp / getEvent

Namespace	DR-GW
Diagram	<p>typeAppGetEvent</p> <p>getEvent</p> <p>requestId</p> <p>result</p> <p>app</p>
Type	typeAppGetEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeAppGetEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , app
Children	app, requestId, result
Instance	<pre><getEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <app>{1,1}</app> </getEvent></pre>

Source

```
<xss:element name="getEvent" type="typeAppGetEvent" />
```

Element typeAppGetEvent / app

Namespace	DR-GW
Diagram	<pre> classDiagram class typeApplication { <<typeApplication>> } class app { <<app>> } typeApplication "1..1" *-- "1..1" app typeApplication "1..1" *-- "1..1" addr typeApplication "1..1" *-- "1..1" alias typeApplication "1..1" *-- "1..1" orgblockId </pre>
Type	typeApplication
Properties	content: complex maxOccurs: 1
Model	addr , alias , orgblockId
Children	addr, alias, orgblockId
Instance	<pre> <app xmlns="DR-GW"> <addr>{1,1}</addr> <alias>{1,1}</alias> <orgblockId>{1,1}</orgblockId> </app> </pre>
Source	<pre><xss:element name="app" type="typeApplication" maxOccurs="1" /></pre>

Element typeApplication / addr

Namespace	DR-GW
Diagram	<pre> classDiagram class typeSubscriberAddress { <<typeSubscriberAddress>> } class addr { <<addr>> } typeSubscriberAddress "1..1" *-- "1..1" addr typeSubscriberAddress "1..1" *-- "1..1" ssi typeSubscriberAddress "1..1" *-- "1..1" tsi </pre>
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre> <addr xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </addr> </pre>
Source	<pre><xss:element name="addr" type="typeSubscriberAddress" /></pre>

Element typeApplication / alias

Namespace	DR-GW
Diagram	<pre> classDiagram class xsnormalizedString { <<xsnormalizedString>> } class alias { <<alias>> } alias *-- xsnormalizedString </pre> <p>Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...</p>
Type	xsnormalizedString
Properties	content: simple

Source	<code><xss:element name="alias" type="xs:normalizedString"/></code>
--------	---

Element typeApplication / orgblockId

Namespace	DR-GW
Diagram	<pre> classDiagram class typeOrganisationBlockId { <<complex type>> <<content>> orgblockId *--> orgblockId orgblockIdSimple *--> orgblockIdSimple } </pre>
Type	typeOrganisationBlockId
Properties	content: complex
Model	orgblockId orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<code><orgblockId xmlns="DR-GW"></code> <code> <orgblockId>{1,1}</orgblockId></code> <code> <orgblockIdSimple>{1,1}</orgblockIdSimple></code> <code></orgblockId></code>
Source	<code><xss:element name="orgblockId" type="typeOrganisationBlockId"/></code>

Element interfaceApp / getListEvent

Namespace	DR-GW
Diagram	<pre> classDiagram class typeEvent { <<extension base>> <<content>> requestId result } class typeAppGetListEvent { <<content>> getListEvent *--> getListEvent app *--> listEnd } </pre>
Type	typeAppGetListEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeAppGetListEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , app* , listEnd{0,1}
Children	app, listEnd, requestId, result
Instance	<code><getListEvent xmlns="DR-GW"></code> <code> <requestId>{0,1}</requestId></code> <code> <result>{0,1}</result></code> <code> <app>{0,unbounded}</app></code> <code> <listEnd>{0,1}</listEnd></code> <code></getListEvent></code>
Source	<code><xss:element name="getListEvent" type="typeAppGetListEvent"/></code>

Element typeAppGetListEvent / app

Namespace	DR-GW
-----------	-------

Diagram	
Type	typeApplication
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Model	addr , alias , orgblockId
Children	addr, alias, orgblockId
Instance	<pre><app xmlns="DR-GW"> <addr>{1,1}</addr> <alias>{1,1}</alias> <orgblockId>{1,1}</orgblockId> </app></pre>
Source	<code><xss:element name="app" type="typeApplication" minOccurs="0" maxOccurs="unbounded" /></code>

Element typeAppGetListEvent / listEnd

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<code><xss:element name="listEnd" type="xs:boolean" minOccurs="0" /></code>

Element drgw / system

Namespace	DR-GW
Diagram	
Type	interfaceSystem
Properties	<p>content: complex</p>
Model	tetraStatesEvent logEvent event
Children	event, logEvent, tetraStatesEvent
Instance	<code><system xmlns="DR-GW"></code>

	<pre><tetraStatesEvent>{1,1}</tetraStatesEvent> <logEvent>{1,1}</logEvent> <event>{1,1}</event> </system></pre>
Source	<code><xss:element name="system" type="interfaceSystem" /></code>

Element interfaceSystem / tetraStatesEvent

Namespace	DR-GW
Diagram	<p>Diagram illustrating the class hierarchy:</p> <ul style="list-style-type: none"> tetraStatesEvent (extension of typeEvent) <ul style="list-style-type: none"> Attributes: requestId, result typeSystemTetraStatesEvent (extension of typeEvent) <ul style="list-style-type: none"> Attributes: tcsState, dxtState, cddconnectionState, cddserverState <p>Indication of the subsystem state updates. Contains current states of all subsystem known by the TCS.</p>
Type	typeSystemTetraStatesEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeSystemTetraStatesEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , tcsState{0,1} , dxtState{0,1} , cddconnectionState{0,1} , cddserverState{0,1}
Children	cddconnectionState, cddserverState, dxtState, requestId, result, tcsState
Instance	<pre><tetraStatesEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <tcsState>{0,1}</tcsState> <dxtState>{0,1}</dxtState> <cddconnectionState>{0,1}</cddconnectionState> <cddserverState>{0,1}</cddserverState> </tetraStatesEvent></pre>
Source	<code><xss:element name="tetraStatesEvent" type="typeSystemTetraStatesEvent" /></code>

Element typeSystemTetraStatesEvent / tcsState

Namespace	DR-GW				
Diagram	<p>Diagram illustrating the association:</p> <ul style="list-style-type: none"> tcsState is associated with typeSystemElementState. <p>Specifies connection, server or unit state.</p>				
Type	typeSystemElementState				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Facets	<table border="1"> <tr> <td>enumeration</td> <td>unknown</td> <td>Unknown state.</td> </tr> </table>	enumeration	unknown	Unknown state.	
enumeration	unknown	Unknown state.			

	enumeration	ok	Connection or server is working.
	enumeration	n_Ok	Connection or server is not working.
Source	<xss:element name="tcsState" type="typeSystemElementState" minOccurs="0"/>		

Element typeSystemTetraStatesEvent / dxtState

Namespace	DR-GW		
Diagram			
Type	typeSystemElementState		
Properties	content: simple minOccurs: 0		
Facets	enumeration unknown enumeration ok enumeration n_Ok		
Source	<xss:element name="dxtState" type="typeSystemElementState" minOccurs="0"/>		

Element typeSystemTetraStatesEvent / cddconnectionState

Namespace	DR-GW		
Diagram			
Type	typeSystemElementState		
Properties	content: simple minOccurs: 0		
Facets	enumeration unknown enumeration ok enumeration n_Ok		
Source	<xss:element name="cddconnectionState" type="typeSystemElementState" minOccurs="0"/>		

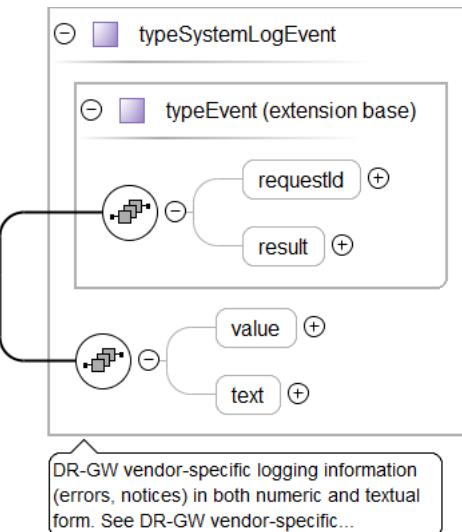
Element typeSystemTetraStatesEvent / cddserverState

Namespace	DR-GW		
Diagram			
Type	typeSystemElementState		
Properties	content: simple minOccurs: 0		
Facets	enumeration unknown enumeration ok enumeration n_Ok		
Source	<xss:element name="cddserverState" type="typeSystemElementState" minOccurs="0"/>		

Element interfaceSystem / logEvent

Namespace	DR-GW
-----------	-------

Diagram



Type	typeSystemLogEvent
------	--------------------

Type hierar-	• typeEvent
chy	• typeSystemLogEvent

Properties	content: complex
------------	------------------

Model	requestId{0,1} , result{0,1} , value{0,1} , text{0,1}
-------	---

Children	requestId, result, text, value
----------	--------------------------------

Instance	<pre> <logEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <value>{0,1}</value> <text>{0,1}</text> </logEvent> </pre>
----------	---

Source	<code><xs:element name="logEvent" type="typeSystemLogEvent" /></code>
--------	---

Element typeSystemLogEvent / value

Namespace	DR-GW
-----------	-------

Diagram	<p>Built-in primitive type. The hexBinary datatype represents arbitrary hex-encoded binary data.</p>
---------	--

Type	xs:hexBinary
------	--------------

Properties	content: simple minOccurs: 0
------------	---------------------------------

Source	<code><xs:element name="value" type="xs:hexBinary" minOccurs="0" /></code>
--------	--

Element typeSystemLogEvent / text

Namespace	DR-GW
-----------	-------

Diagram	<p>Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...</p>
---------	--

Type	xs:normalizedString
------	---------------------

Properties	content: simple minOccurs: 0
------------	---------------------------------

Source

```
<xss:element name="text" type="xs:normalizedString" minOccurs="0"/>
```

Element interfaceSystem / event

Namespace	DR-GW
Diagram	<p>DR-GW vendor-specific system information in both numeric and textual form. See DR-GW vendor-specific documentation for...</p>
Type	typeSystemEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeSystemEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , value{0,1} , text{0,1}
Children	requestId, result, text, value
Instance	<pre><event xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <value>{0,1}</value> <text>{0,1}</text> </event></pre>
Source	<pre><xss:element name="event" type="typeSystemEvent" /></pre>

Element typeSystemEvent / value

Namespace	DR-GW				
Diagram	<p>Built-in primitive type. The hexBinary datatype represents arbitrary hex-encoded binary data.</p>				
Type	xs:hexBinary				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<pre><xss:element name="value" type="xs:hexBinary" minOccurs="0"/></pre>				

Element typeSystemEvent / text

Namespace	DR-GW
Diagram	<p>Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...</p>

Type	xs:normalizedString
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<xss:element name="text" type="xs:normalizedString" minOccurs="0"/>

Element typeCallRequest / action

Namespace	DR-GW																					
Diagram	<pre> classDiagram class action class typeActionRequest { <<All possible call actions.>> } action < -- typeActionRequest </pre>																					
Type	typeActionRequest																					
Properties	content: simple																					
Facets	<table> <tr> <td>enumeration</td> <td>setup</td> <td>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.</td> </tr> <tr> <td>enumeration</td> <td>connect</td> <td>This method is used to connect an incoming call.</td> </tr> <tr> <td>enumeration</td> <td>hold</td> <td>This method requests to put an individual call to hold.</td> </tr> <tr> <td>enumeration</td> <td>unhold</td> <td>This method is a request for resuming an individual call from hold.</td> </tr> <tr> <td>enumeration</td> <td>disconnect</td> <td>This method is used to disconnect a call.</td> </tr> <tr> <td>enumeration</td> <td>transfer</td> <td>This method is used to transfer an individual call to a new recipient.</td> </tr> <tr> <td>enumeration</td> <td>releasecall</td> <td>This method is used to release radio subscriber's individual call.</td> </tr> </table>	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.
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	<xss:element name="action" type="typeActionRequest"/>																					

Element typeCallRequest / attributes

Namespace	DR-GW
Diagram	<pre> classDiagram class attributes class typeCallAttributes { <<Contains all attributes of the TETRA voice call.>> } attributes < -- typeCallAttributes </pre>
Type	typeCallAttributes

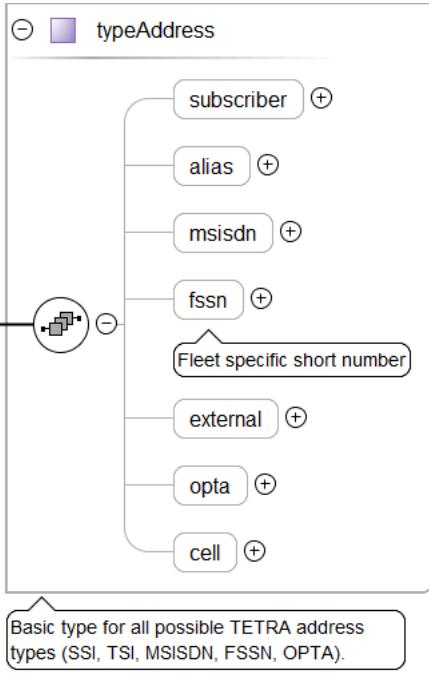
Properties	content: complex minOccurs: 0
Model	hook{0,1} , mode{0,1} , commtype{0,1} , priority{0,1} , encryption{0,1} , ambienceListen{0,1} , req2speak{0,1} , demandPriority{0,1}
Children	ambienceListen, commtype, demandPriority, encryption, hook, mode, priority, req2speak
Instance	<pre><attributes xmlns="DR-GW"> <hook>{0,1}</hook> <mode>{0,1}</mode> <commtype>{0,1}</commtype> <priority>{0,1}</priority> <encryption>{0,1}</encryption> <ambienceListen>{0,1}</ambienceListen> <req2speak>{0,1}</req2speak> <demandPriority>{0,1}</demandPriority> </attributes></pre>
Source	<code><xss:element name="attributes" type="typeCallAttributes" minOccurs="0"/></code>

Element typeCallRequest / callingParty

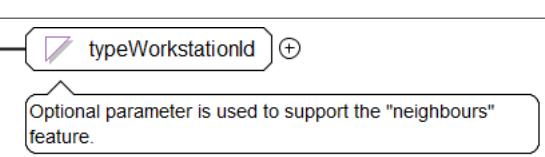
Namespace	DR-GW
Diagram	<p>The diagram shows a class named 'typeAddress' with several associations. One association is labeled 'callingParty' with a multiplicity of 0..1. Another association is labeled with a small square icon. Below the class, a note states: 'Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA)'.</p>
Type	typeAddress
Properties	content: complex minOccurs: 0
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<pre><callingParty xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </callingParty></pre>
Source	<code><xss:element name="callingParty" type="typeAddress" minOccurs="0"/></code>

Element typeCallRequest / calledParty

Namespace	DR-GW
-----------	-------

Diagram	 <p>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</p>				
Type	typeAddress				
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> <tr> <td style="padding: 2px;">minOccurs:</td><td style="padding: 2px;">0</td></tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}				
Children	alias, cell, external, fssn, msisdn, opta, subscriber				
Instance	<pre><calledParty xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </calledParty></pre>				
Source	<code><xss:element name="calledParty" type="typeAddress" minOccurs="0" /></code>				

Element typeCallRequest / workstationId

Namespace	DR-GW				
Diagram	 <p>Optional parameter is used to support the "neighbours" feature.</p>				
Type	typeWorkstationId				
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">simple</td></tr> <tr> <td style="padding: 2px;">minOccurs:</td><td style="padding: 2px;">0</td></tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<code><xss:element name="workstationId" type="typeWorkstationId" minOccurs="0" /></code>				

Element check

Namespace	DR-GW
-----------	-------

Diagram	<p>To enable the DF-Client to check connectivity to DR-GW the client may use session check. The check requires the http...</p>
Type	typeSessionCheck
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeSessionCheck
Properties	content: complex
Model	requestId , clientid
Children	clientid, requestId
Instance	<pre><check xmlns="DR-GW"> <requestId>{1,1}</requestId> <clientid>{1,1}</clientid> </check></pre>
Source	<code><xs:element name="check" type="typeSessionCheck" /></code>

Element typeSessionLogoutEvent / reason

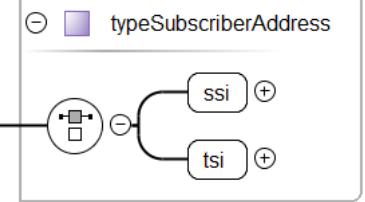
Namespace	DR-GW
Diagram	<p>Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...</p>
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	<code><xs:element name="reason" type="xs:unsignedLong" minOccurs="0" /></code>

Element typeSdsValidity / value

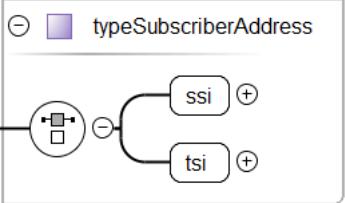
Namespace	DR-GW
Diagram	<p>Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...</p>
Type	xs:unsignedLong
Properties	content: simple
Source	<code><xs:element name="value" type="xs:unsignedLong" /></code>

Element typeGroupGetCombinationsEvent / group

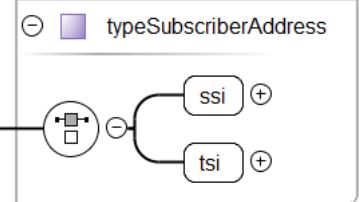
Namespace	DR-GW
-----------	-------

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group></pre>
Source	<code><xss:element name="group" type="typeSubscriberAddress"/></code>

Element typeGroupGetCombinationsEvent / baseGroup

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><baseGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </baseGroup></pre>
Source	<code><xss:element name="baseGroup" type="typeSubscriberAddress"/></code>

Element typeGroupGetCombinationsEvent / constitGroup

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex maxOccurs: 7
Model	ssi tsi
Children	ssi, tsi
Instance	<pre><constitGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </constitGroup></pre>
Source	<code><xss:element name="constitGroup" type="typeSubscriberAddress" maxOccurs="7"/></code>

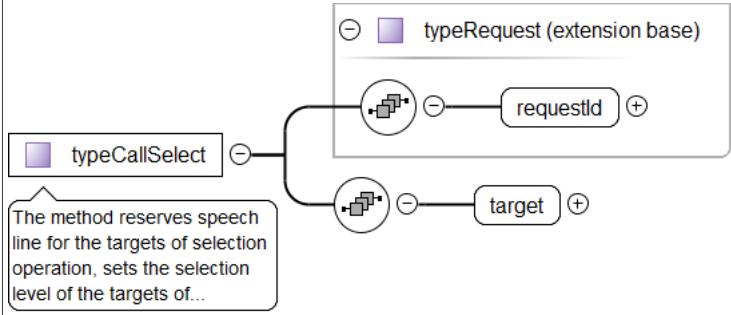
Complex Type(s)

Complex Type interfaceCall

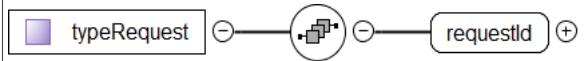
Namespace	DR-GW
Annotations	DR-GW-Call. Use for call control/ call monitoring. This is the only element, that can be used via both SIP/SOAP. When used via SOAP it can only be used for call event monitoring.
Diagram	<pre> classDiagram class interfaceCall { <<DR-GW-Call. Use for call control/ call monitoring. This is the only element, that can be used via both SIP/SOAP. When...>> } interfaceCall "1" -- "1" choice choice "1" -- "1" select choice "1" -- "1" request choice "1" -- "1" pttRequest choice "1" -- "1" keyExchange choice "1" -- "1" response choice "1" -- "1" selectEvent choice "1" -- "1" event choice "1" -- "1" pttEvent choice "1" -- "1" unitInEmergencyEvent choice "1" -- "1" keyExchangeEvent </pre>
Used by	Element drgw/call
Model	select request pttRequest keyExchange response selectEvent event pttEvent unitInEmergencyEvent keyExchangeEvent
Children	event, keyExchange, keyExchangeEvent, pttEvent, pttRequest, request, response, select, selectEvent, unitInEmergencyEvent
Source	<pre> <xs:complexType name="interfaceCall"> <xs:annotation> <xs:documentation>DR-GW-Call. Use for call control/ call monitoring. This is the only element, that can be used via both SIP/SOAP. When used via SOAP it can only be used for call event monitoring.</xs:documentation> </xs:annotation> <xs:choice> <xs:element name="select" type="typeCallSelect"/> <xs:element name="request" type="typeCallEvent"/> <xs:element name="pttRequest" type="typeCallPTTRequest"/> <xs:element name="keyExchange" type="typeCallKeyExchange"/> <xs:element name="response" type="typeResponse"/> <xs:element name="selectEvent" type="typeCallSelectEvent"/> <xs:element name="event" type="typeCallEvent"/> <xs:element name="pttEvent" type="typeCallPTTEvent"/> <xs:element name="unitInEmergencyEvent" type="typeCallUnitInEmergencyEvent"/> <xs:element name="keyExchangeEvent" type="typeCallKeyExchangeEvent"/> </xs:choice> </xs:complexType> </pre>

Complex Type typeCallSelect

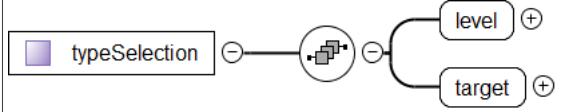
Namespace	DR-GW
Annotations	The method reserves speech line for the targets of selection operation, sets the selection level of the targets of selection operation to given values.

Diagram	
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeCallSelect
Used by	Element interfaceCall/select
Model	requestId , target
Children	requestId, target
Source	<pre><xs:complexType name="typeCallSelect"> <xs:annotation> <xs:documentation>The method reserves speech line for the targets of selection operation, sets the selection level of the targets of selection operation to given values.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="target" type="typeSelection"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeRequest

Namespace	DR-GW
Diagram	
Used by	Complex Types typeAppGet, typeAppGetList, typeCallKeyExchange, typeCallPTTRequest, typeCallRequest, typeCallSelect, typeGroupAddCombination, typeGroupAddRadioMember, typeGroupGet, typeGroupGetAppMembers, typeGroupGetCombinations, typeGroupGetList, typeGroupGetRadioMembers, typeGroupRemoveCombination, typeGroupRemoveRadioMember, typeGroupSubscribeData, typeGroupTrack, typeOrgGet, typeOrgGetList, typeRadioChangeOpta, typeRadioDisable, typeRadioEnable, typeRadioGet, typeRadioGetGroups, typeRadioGetList, typeRadioTrack, typeSdsSend, typeSdsSendReport, typeSessionCheck, typeSessionLogin, typeSessionLogout, typeSessionSupervise, typeStatusSend
Model	requestId
Children	requestId
Source	<pre><xs:complexType name="typeRequest"> <xs:sequence> <xs:element name="requestId" type="xs:unsignedLong"/> </xs:sequence> </xs:complexType></pre>

Complex Type typeSelection

Namespace	DR-GW
Annotations	
Diagram	
Used by	Elements typeCallSelect/target, typeCallSelectEvent/target
Model	level , target

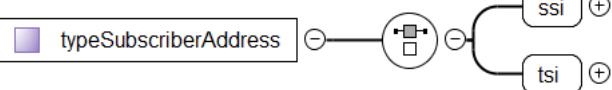
Children	level, target
Source	<pre><xs:complexType name="typeSelection"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="level" type="typeSelectionLevel"/> <xs:element name="target" type="typeAddress"/> </xs:sequence> </xs:complexType></pre>

Complex Type typeAddress

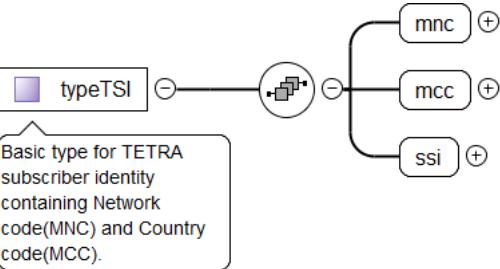
Namespace	DR-GW
Annotations	Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).
Diagram	<pre> classDiagram typeAddress < -- subscriber typeAddress < -- alias typeAddress < -- msisdn typeAddress < -- fssn typeAddress < -- external typeAddress < -- opta typeAddress < -- cell typeAddress -- "0..1" *--> subscriber typeAddress -- "0..1" *--> alias typeAddress -- "0..1" *--> msisdn typeAddress -- "0..1" *--> fssn typeAddress -- "0..1" *--> external typeAddress -- "0..1" *--> opta typeAddress -- "0..1" *--> cell </pre> <p>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</p>
Used by	Elements typeCallEvent/calledParty, typeCallEvent/callingParty, typeCallPTTRequest/talkingParty, typeCallRequest/calledParty, typeCallRequest/callingParty, typeCallUnitInEmergencyEvent/unitInEmg, typeSds/forward, typeSds/source, typeSds/target, typeSdsReportEvent/source, typeSdsReportEvent/target, typeSdsSendReport/target, typeSelection/target, typeStatus/source, typeStatus/target, typeTxGranted/talkingParty
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Source	<pre><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="typeSubscriberAddress" minOccurs="0"/> <xs:element name="alias" type="xs:normalizedString" minOccurs="0"/> <xs:element name="msisdn" type="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="typeExternal" minOccurs="0"/> <xs:element name="opta" type="typeOpta" minOccurs="0"/> <xs:element name="cell" type="xs:short" minOccurs="0"/> </xs:sequence> </xs:complexType></pre>

Complex Type typeSubscriberAddress

Namespace	DR-GW
Annotations	

Diagram	
Used by	Elements typeAddress/subscriber, typeAppGet/app, typeApplication/addr, typeCallUnitInEmergencyEvent/group, typeGroup/addr, typeGroupAddCombination/baseGroup, typeGroupAddCombination/group, typeGroupAddCombinationEvent/baseGroup, typeGroupAddCombinationEvent/group, typeGroupAddRadioMember/group, typeGroupAddRadioMember/radio, typeGroupAddRadioMemberEvent/group, typeGroupAddRadioMemberEvent/radio, typeGroupAppMemberEvent/app, typeGroupCombinationEvent/baseGroup, typeGroupCombinationEvent/constitGroup, typeGroupCombinationEvent/group, typeGroupDataSubscription/addr, typeGroupGet/group, typeGroupGetAppMembers/group, typeGroupGetAppMembersEvent/app, typeGroupGetCombinations/group, typeGroupGetCombinationsEvent/baseGroup, typeGroupGetCombinationsEvent/constitGroup, typeGroupGetCombinationsEvent/group, typeGroupGetRadioMembers/group, typeGroupGetRadioMembersEvent/group, typeGroupGetRadioMembersEvent/radio, typeGroupGetRadioMemberEvent/group, typeGroupRadioMemberEvent/radio, typeGroupRemoveCombination/baseGroup, typeGroupRemoveCombination/group, typeGroupRemoveCombinationEvent/baseGroup, typeGroupRemoveCombinationEvent/group, typeGroupRemoveRadioMember/group, typeGroupRemoveRadioMemberEvent/radio, typeGroupRemoveRadioMemberEvent/group, typeGroupTrack/group, typeGroupTrackSubscriptionEvent/group, typeRadio/issi, typeRadioChangeOpta/radio, typeRadioChangeOptaEvent/radio, typeRadioDisable/radio, typeRadioEnable/radio, typeRadioEnableDisableEvent/radio, typeRadioGet/radio, typeRadioGetGroupsEvent/radio, typeRadioGroupSelection/group, typeRadioGroupsEvent/deletedGroup, typeRadioGroupsEvent/radio, typeRadioTrack/radio, typeRadioTrackSubscriptionEvent/radio, typeRadioTrackingData/callParty, typeRadioTrackingData/radio, typeSds/e2eegroup
Model	ssi tsi
Children	ssi, tsi
Source	<pre><xs:complexType name="typeSubscriberAddress"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:choice> <xs:element name="ssi" type="xs:unsignedLong"/> <xs:element name="tsi" type="typeTSI"/> </xs:choice> </xs:complexType></pre>

Complex Type typeTSI

Namespace	DR-GW
Annotations	Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC).
Diagram	 <p>Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC).</p>
Used by	Element typeSubscriberAddress/tsi
Model	mnc , mcc , ssi
Children	mcc, mnc, ssi
Source	<pre><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></pre>

Complex Type typeExternal

Namespace	DR-GW
Annotations	External number consisting of Gateway number + DialString
Diagram	<pre> classDiagram typeExternal < -- typeEvent typeExternal >-- gatewayNumber typeExternal >-- number gatewayNumber >--> number note over typeExternal, gatewayNumber, number: External number consisting of Gateway number + DialString </pre>
Used by	Element typeAddress/external
Model	gatewayNumber , number
Children	gatewayNumber, number
Source	<pre> <xs:complexType name="typeExternal"> <xs:annotation> <xs:documentation>External number consisting of Gateway number + DialString</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="gatewayNumber" type="xs:unsignedLong"/> <xs:element name="number" type="typeDialString"/> </xs:sequence> </xs:complexType> </pre>

Complex Type typeCallEvent

Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeEvent < -- typeCallEvent typeCallEvent >-- requestId typeCallEvent >-- result typeCallEvent >-- tetraCallId typeCallEvent >-- action typeCallEvent >-- attributes typeCallEvent >-- callingParty typeCallEvent >-- calledParty typeCallEvent >-- disconnectCause note over typeEvent, typeCallEvent: typeEvent (extension base) </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeCallEvent
Used by	Elements interfaceCall/event, interfaceCall/request
Model	requestId{0,1} , result{0,1} , tetraCallId{0,1} , action , attributes{0,1} , callingParty{0,1} , calledParty{0,1} , disconnectCause{0,1}
Children	action, attributes, calledParty, callingParty, disconnectCause, requestId, result, tetraCallId
Source	<pre> <xs:complexType name="typeCallEvent"> <xs:annotation> <xs:documentation></xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="requestId" type="xs:string" minOccurs="0" maxOccurs="1"/> <xs:element name="result" type="xs:string" minOccurs="0" maxOccurs="1"/> <xs:element name="tetraCallId" type="xs:string" minOccurs="0" maxOccurs="1"/> <xs:element name="action" type="xs:string" minOccurs="0" maxOccurs="1"/> <xs:element name="attributes" type="xs:string" minOccurs="0" maxOccurs="1"/> <xs:element name="callingParty" type="xs:string" minOccurs="0" maxOccurs="1"/> <xs:element name="calledParty" type="xs:string" minOccurs="0" maxOccurs="1"/> <xs:element name="disconnectCause" type="xs:string" minOccurs="0" maxOccurs="1"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

```

<xs:extension base="typeEvent">
  <xs:sequence>
    <xs:element name="tetraCallId" type="xs:unsignedLong" minOccurs="0"/>
    <xs:element name="action" type="typeActionEvent"/>
    <xs:element name="attributes" type="typeCallAttributes" minOccurs="0"/>
    <xs:element name="callingParty" type="typeAddress" minOccurs="0"/>
    <xs:element name="calledParty" type="typeAddress" minOccurs="0"/>
    <xs:element name="disconnectCause" type="typeDisconnectCause" minOccurs="0"/>
  </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

Complex Type typeEvent

Namespace	DR-GW
Diagram	<pre> classDiagram class typeEvent class requestId class result typeEvent "0..1" -- "0..1" requestId : requestId typeEvent "0..1" -- "0..1" result : result </pre>
Used by	<p>Complex Types</p> <p>typeAppGetEvent, typeAppGetListEvent, typeCallEvent, typeCallKeyExchangeEvent, typeCallPT-TEvent, typeCallSelectEvent, typeCallUnitInEmergencyEvent, typeGroupAddCombinationEvent, typeGroupAddRadioMemberEvent, typeGroupAppMemberEvent, typeGroupCombinationEvent, typeGroupEvent, typeGroupGetAppMembersEvent, typeGroupGetCombinationsEvent, typeGroupGetEvent, typeGroupGetListEvent, typeGroupGetRadioMembersEvent, typeGroupRadioMemberEvent, typeGroupRemoveCombinationEvent, typeGroupRemoveRadioMemberEvent, typeGroupSubscribeDataEvent, typeGroupTrackSubscriptionEvent, typeOrgEvent, typeOrgGetEvent, typeOrgGetListEvent, typeRadioChangeOptaEvent, typeRadioEnableDisableEvent, typeRadioEvent, typeRadioGetEvent, typeRadioGetGroupsEvent, typeRadioGetListEvent, typeRadioGroupsEvent, typeRadioTrackEvent, typeRadioTrackSubscriptionEvent, typeSdsReceiveEvent, typeSdsReportEvent, typeSdsSendEvent, typeSessionLoginEvent, typeSessionLogoutEvent, typeSessionSuperviseEvent, typeStatusReceiveEvent, typeStatusSendEvent, typeSystemEvent, typeSystemLogEvent, typeSystemTetraStateEvent</p>
Model	requestId{0,1} , result{0,1}
Children	requestId, result
Source	<pre> <xs:complexType name="typeEvent"> <xs:sequence> <xs:element name="requestId" type="xs:unsignedLong" minOccurs="0"/> <xs:element name="result" type="typeResult" minOccurs="0"/> </xs:sequence> </xs:complexType> </pre>

Complex Type typeResult

Namespace	DR-GW
Annotations	Common result values used in every response and optional specific subsystem result codes.
Diagram	<pre> classDiagram class typeResult class responseCode class sourceSystem class result typeResult "0..1" -- "0..1" responseCode : responseCode typeResult "0..1" -- "0..1" sourceSystem : sourceSystem typeResult "0..1" -- "0..1" result : result </pre>
Used by	Elements typeEvent/result, typeResponse/result
Model	responseCode , sourceSystem{0,1} , result{0,1}
Children	responseCode, result, sourceSystem
Source	<pre> <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="typeResponseCode"/> <xs:element name="sourceSystem" type="typeSourceSystem" minOccurs="0"/> <xs:element name="result" type="xs:unsignedLong" minOccurs="0"/> </xs:sequence> </xs:complexType> </pre>

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

Complex Type typeCallAttributes

Namespace	DR-GW
Annotations	Contains all attributes of the TETRA voice call.
Diagram	<p>The diagram illustrates the structure of the typeCallAttributes complex type. It consists of a central node labeled "typeCallAttributes" with a multiplicity of 0..1. Eight arrows point from this central node to individual attributes, each preceded by a plus sign (+) indicating they are optional. The attributes are: hook, mode, commtype, priority, encryption, ambienceListen, req2speak, and demandPriority.</p> <p>Contains all attributes of the TETRA voice call.</p>
Used by	Elements typeCallEvent/attributes, typeCallPTTRequest/attributes, typeCallRequest/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	<pre><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></pre>

Complex Type typeDisconnectCause

Namespace	DR-GW
Diagram	<p>The diagram illustrates the structure of the typeDisconnectCause complex type. It consists of a central node labeled "typeDisconnectCause" with a multiplicity of 0..1. Three arrows point from this central node to individual attributes, each preceded by a plus sign (+) indicating they are optional. The attributes are: protocol, code, and text.</p> <p>Value according to DR-GW-Reason header in DR-GW-Interface specification. It should only be present if action is...</p> <p>Optional textual representation of the cause.</p>
Used by	Element typeCallEvent/disconnectCause
Model	protocol , code , text{0,1}
Children	code, protocol, text

Source	<pre> <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> </pre>
--------	---

Complex Type typeCallPTTRequest

Namespace	DR-GW
Annotations	DR-GW-Call PTTRequest. Only "DemandTx" and "CeaseTx" actions only.
Diagram	<pre> classDiagram typeRequest --> typeCallPTTRequest typeRequest { -requestId -action -attributes -talkingParty -workstationId } typeCallPTTRequest { +DR-GW-Call PTTRequest. Only "DemandTx" and "CeaseTx" actions only. } </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeCallPTTRequest
Used by	Element interfaceCall/pttRequest
Model	requestId , action , attributes{0,1} , talkingParty{0,1} , workstationId{0,1}
Children	action, attributes, requestId, talkingParty, workstationId
Source	<pre> <xs:complexType name="typeCallPTTRequest"> <xs:annotation> <xs:documentation>DR-GW-Call PTTRequest. Only "DemandTx" and "CeaseTx" actions only.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="action" type="typeActionPTTRequest"/> <xs:element name="attributes" type="typeCallAttributes" minOccurs="0"/> <xs:element name="talkingParty" type="typeAddress" minOccurs="0"/> <xs:element name="workstationId" type="typeWorkstationId" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeCallKeyExchange

Namespace	DR-GW
Annotations	For triggering the group key exchange. Key exchange events are sent in Call_KeyXEvent.
Diagram	<pre> classDiagram typeRequest < -- typeCallKeyExchange typeRequest "0..1" --> requestId : String typeRequest "0..1" --> action : typeKeyExchangeAction typeCallKeyExchange "0..1" --> requestId : String typeCallKeyExchange "0..1" --> action : typeKeyExchangeAction </pre> <p>For triggering the group key exchange. Key exchange events are sent in Call_KeyXEvent.</p>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeCallKeyExchange
Used by	Element interfaceCall/keyExchange
Model	requestId , action
Children	action, requestId
Source	<pre> <xs:complexType name="typeCallKeyExchange"> <xs:annotation> <xs:documentation>For triggering the group key exchange. Key exchange events are sent in Call_KeyXEvent.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="action" type="typeKeyExchangeAction"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeResponse

Namespace	DR-GW
Annotations	Response contains result of execution of any method.
Diagram	<pre> classDiagram typeResponse "1..1" --> requestId : String typeResponse "1..1" --> result : typeResult </pre> <p>Response contains result of execution of any method.</p>
Used by	Elements interfaceApp/response, interfaceCall/response, interfaceGroup/response, interfaceOrg/response, interfaceRadio/response, interfaceSds/response, interfaceSession/response, interfaceStatus/response
Model	requestId , result
Children	requestId, result
Source	<pre> <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="typeResult"/> </xs:sequence> </xs:complexType> </pre>

Complex Type typeCallSelectEvent

Namespace	DR-GW
Annotations	The event informs about the actual state of the selection requested before using the "select" request.

Diagram	<p>The event informs about the actual state of the selection requested before using the "select" request.</p>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeCallSelectEvent
Used by	Element interfaceCall/selectEvent
Model	requestId{0,1} , result{0,1} , target
Children	requestId, result, target
Source	<pre><xss:complexType name="typeCallSelectEvent"> <xss:annotation> <xss:documentation>The event informs about the actual state of the selection requested before using the "select" request.</xss:documentation> </xss:annotation> <xss:complexContent> <xss:extension base="typeEvent"> <xss:sequence> <xss:element name="target" type="typeSelection"/> </xss:sequence> </xss:extension> </xss:complexContent> </xss:complexType></pre>

Complex Type typeCallPTTEvent

Namespace	DR-GW
Annotations	
Diagram	<p>This event is used to inform that transmission is ceased and nobody has the speech item.</p> <p>This event is used to inform that the call is temporarily paused e.g. if radio subscriber has roamed to a new cell and...</p>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeCallPTTEvent

Used by	Element	interfaceCall/pttEvent
Model		requestId{0,1} , result{0,1} , tetraCallId{0,1} , (granted ceased wait)
Children		ceased, granted, requestId, result, tetraCallId, wait
Source		<pre> <xs:complexType name="typeCallPTTEvent"> <xs:annotation> <xs:documentation> </xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="tetraCallId" type="xs:unsignedLong" minOccurs="0"/> <xs:choice> <xs:element name="granted" type="typeTxGranted"/> <xs:element name="ceased" type="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="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> </pre>

Complex Type typeTxGranted

Namespace	DR-GW
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	<p>The diagram illustrates the structure of the typeTxGranted complex type. It features a central class box labeled "typeTxGranted" with a purple square icon. A line with a hollow circle at the end connects it to a rounded rectangle containing the text: "This event is used to inform of granted transmission request. This event is called when the transmission is granted to...". From the "typeTxGranted" box, a line with a solid circle at the end connects to a sequence of five boxes: "txGrant" (with a plus sign), "talkingParty" (with a plus sign), "encryption" (with a plus sign), "txPriority" (with a plus sign), and "txInterrupt" (with a plus sign). A line with a solid circle at the end connects "txInterrupt" to a callout box containing the text: "Defines whether previous speaker's speech item was interrupted by this speech item. Valid only when txGrant is...". Another line with a solid circle at the end connects "txRepeat" (with a plus sign) to a callout box containing the text: "Timer to repeat the PTT. Units are seconds. Always suggested by the DF-gateway. Only valid when txGrant=granted.". Finally, a line with a solid circle at the end connects "workstationId" (with a plus sign) to a callout box containing the text: "Id of the currently speaking workstation, used for \"neighbours\" feature. Only valid when txGrant=granted and when...".</p>
Used by	Element
	typeCallPTTEvent/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	<pre> <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="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></pre>

Complex Type typeEmpty

Namespace	DR-GW
Annotations	Explicit type specification for elements that shall be empty.
Diagram	<p>The diagram shows a rectangular box containing the text "typeEmpty". A curved arrow originates from the bottom-left of this box and points to the explanatory text "Explicit type specification for elements that shall be empty." located below it.</p>
Used by	Elements typeCallPTTEvent/ceased, typeCallPTTEvent/wait
Source	<pre> <xs:complexType name="typeEmpty"> <xs:annotation> <xs:documentation>Explicit type specification for elements that shall be empty.</xs:documentation> </xs:annotation> </xs:complexType></pre>

Complex Type typeCallUnitInEmergencyEvent

Namespace	DR-GW
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	<pre> classDiagram typeEvent { requestId result } typeCallUnitInEmergencyEvent { group tetraCallId unitInEmg unitInEmgType emgInfo tstamp } typeEvent < -- typeCallUnitInEmergencyEvent note over typeCallUnitInEmergencyEvent: This event is used to inform DF-Client that a subscriber is in emergency state during an emergency group call including... </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeCallUnitInEmergencyEvent
Used by	Element interfaceCall/unitInEmergencyEvent
Model	requestId{0,1} , result{0,1} , group , tetraCallId{0,1} , unitInEmg , unitInEmgType , emgInfo , tstamp
Children	emgInfo, group, requestId, result, tetraCallId, tstamp, unitInEmg, unitInEmgType
Source	<pre> <xs:complexType name="typeCallUnitInEmergencyEvent"> <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:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> <xs:element name="tetraCallId" type="xs:unsignedLong" minOccurs="0"/> <xs:element name="unitInEmg" type="typeAddress"/> <xs:element name="unitInEmgType" type="typeUnitInEmergencyType"/> <xs:element name="emgInfo" type="typeEmergencyInfo"/> <xs:element name="tstamp" type="xs:dateTime"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

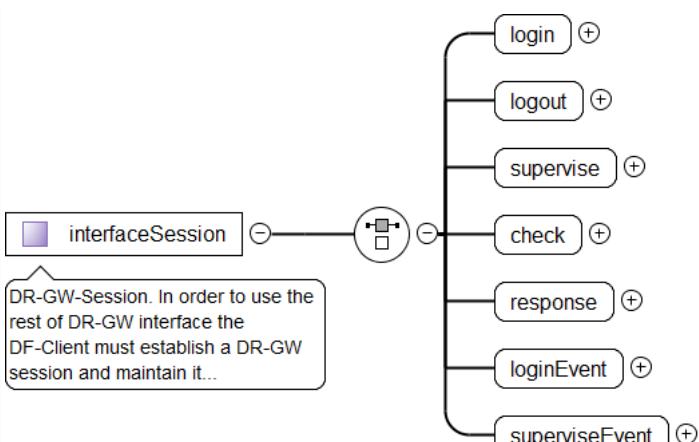
Complex Type typeCallKeyExchangeEvent

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

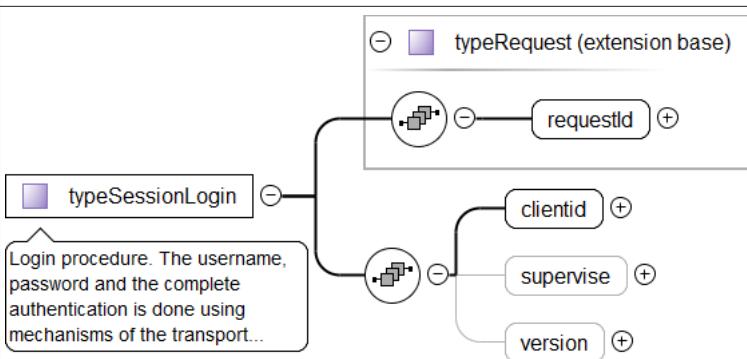
Diagram	<pre> classDiagram typeEvent { requestId result } typeCallKeyExchangeEvent { state code choice { priority interaction text } } typeEvent < -- typeCallKeyExchangeEvent state --> "Represents the current key exchange state." interaction --> "If user interaction is required, then the message box should be visible until the interaction is made, otherwise should..." </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeCallKeyExchangeEvent
Used by	Element interfaceCall/keyExchangeEvent
Model	requestId{0,1} , result{0,1} , state , code , ((priority{0,1} , interaction{0,1} , text) tone)
Children	code, interaction, priority, requestId, result, state, text, tone
Source	<pre> <xs:complexType name="typeCallKeyExchangeEvent"> <xs:annotation> <xs:documentation>Events regarding key exchange. Contains the complete key exchange information from the BOS-Simcard.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="state" type="typeKeyExchangeState"> <xs:annotation> <xs:documentation>Represents the current key exchange state.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="code" type="typeKeyExchangeCode" /> <xs:choice> <xs:sequence> <xs:element name="priority" type="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="typeKeyExchangeText" /> </xs:sequence> <xs:element name="tone" type="xs:boolean" fixed="true" /> </xs:choice> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type interfaceSession

Namespace	DR-GW
Annotations	DR-GW-Session. In order to use the rest of DR-GW interface the DF-Client must establish a DR-GW session and maintain it using supervise in a timely manner. Use only via SOAP.

Diagram	
Used by	Element drgw/session
Model	login logout supervise check response loginEvent superviseEvent
Children	check, login, loginEvent, logout, response, supervise, superviseEvent
Source	<pre><xs:complexType name="interfaceSession"> <xs:annotation> <xs:documentation>DR-GW-Session. In order to use the rest of DR-GW interface the DF-Client must establish a DR-GW session and maintain it using supervise in a timely manner. Use only via SOAP.</xs:documentation> </xs:annotation> <xs:choice> <xs:element name="login" type="typeSessionLogin"/> <xs:element name="logout" type="typeSessionLogout"/> <xs:element name="supervise" type="typeSessionSupervise"/> <xs:element name="check" type="typeSessionCheck"/> <xs:element name="response" type="typeResponse"/> <xs:element name="loginEvent" type="typeSessionLoginEvent"/> <xs:element name="superviseEvent" type="typeSessionSuperviseEvent"/> </xs:choice> </xs:complexType></pre>

Complex Type typeSessionLogin

Namespace	DR-GW
Annotations	Login procedure. The username, password and the complete authentication is done using mechanisms of the transport protocol, digest access authentication on HTTP.
Diagram	
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeSessionLogin
Used by	Element interfaceSession/login
Model	requestId , clientid , supervise{0,1} , version{0,1}
Children	clientid, requestId, supervise, version
Source	<pre><xs:complexType name="typeSessionLogin"> <xs:annotation></pre>

```

<xs:documentation>Login procedure. The username, password and the complete authentication
is done using mechanisms of the transport protocol, digest access authentication on HTTP.</
xs:documentation>
</xs:annotation>
<xs:complexContent>
  <xs:extension base="typeRequest">
    <xs:sequence>
      <xs:element name="clientId" type="xs:string"/>
      <xs:element name="supervise" type="typeSuperviseTimeout" default="60" minOccurs="0"/>
      <xs:element name="version" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>

```

Complex Type typeSessionLogout

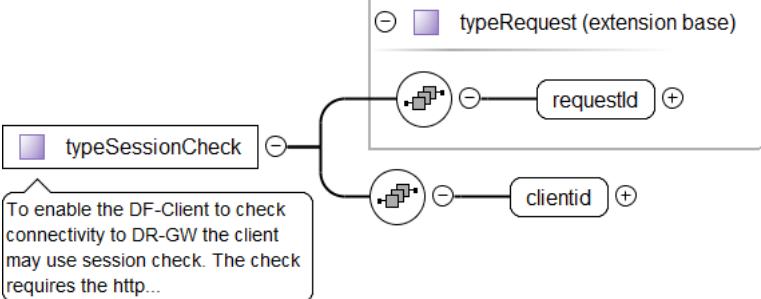
Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeRequest < -- typeSessionLogout typeRequest { <> clientId : string <> supervise : typeSuperviseTimeout <> version : string } typeSessionLogout { <> requestId : string } </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeSessionLogout
Used by	Element interfaceSession/logout
Model	requestId
Children	requestId
Source	<pre> <xs:complexType name="typeSessionLogout"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeSessionSupervise

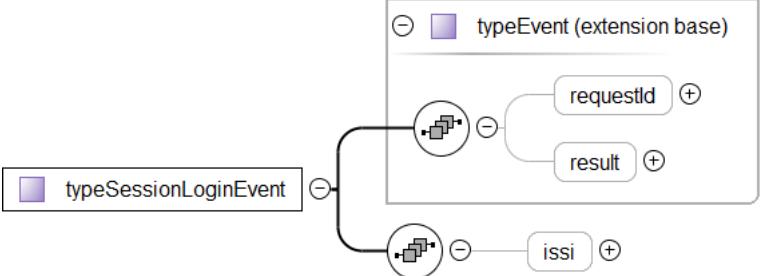
Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeRequest < -- typeSessionSupervise typeRequest { <> clientId : string <> supervise : typeSuperviseTimeout <> version : string } typeSessionSupervise { <> requestId : string } </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeSessionSupervise
Used by	Element interfaceSession/supervise
Model	requestId
Children	requestId
Source	<pre> <xs:complexType name="typeSessionSupervise"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

<pre></xs:complexType></pre>

Complex Type typeSessionCheck

Namespace	DR-GW
Annotations	To enable the DF-Client to check connectivity to DR-GW the client may use session check. The check requires the http authentication as it would be needed for a session login. Once the session check is OK, the client should be able to login later on. There is no resource allocation associated with the session check.
Diagram	 <pre> classDiagram typeRequest < -- typeSessionCheck typeRequest { -requestId -clientid } typeSessionCheck { note "To enable the DF-Client to check connectivity to DR-GW the client may use session check. The check requires the http..." } </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeSessionCheck
Used by	Elements check, interfaceSession/check
Model	requestId , clientid
Children	clientid, requestId
Source	<pre> <xs:complexType name="typeSessionCheck"> <xs:annotation> <xs:documentation>To enable the DF-Client to check connectivity to DR-GW the client may use session check. The check requires the http authentication as it would be needed for a session login. Once the session check is OK, the client should be able to login later on. There is no resource allocation associated with the session check.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="clientid" type="xs:string"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeSessionLoginEvent

Namespace	DR-GW
Annotations	
Diagram	 <pre> classDiagram typeEvent < -- typeSessionLoginEvent typeEvent { -requestId -result } typeSessionLoginEvent { -issi } </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeSessionLoginEvent
Used by	Element interfaceSession/loginEvent

Model	requestId{0,1} , result{0,1} , issi{0,1}
Children	issi, requestId, result
Source	<pre><xs:complexType name="typeSessionLoginEvent"> <xs:annotation> <xs:documentation> </xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="issi" type="xs:string" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeSessionSuperviseEvent

Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeEvent < -- typeSessionSuperviseEvent typeEvent { +requestId +result } </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeSessionSuperviseEvent
Used by	Element interfaceSession/superviseEvent
Model	requestId{0,1} , result{0,1}
Children	requestId, result
Source	<pre><xs:complexType name="typeSessionSuperviseEvent"> <xs:annotation> <xs:documentation> </xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent" /> </xs:complexContent> </xs:complexType></pre>

Complex Type interfaceSds

Namespace	DR-GW
Annotations	DR-GW-Sds element. Use to send/receive SDS messages. Use only via SOAP.
Diagram	<pre> interfaceSds { +send +sendReport +response +sendEvent +receiveEvent +reportEvent } </pre> <p>DR-GW-Sds element. Use to send/receive SDS messages. Use only via SOAP.</p>
Used by	Element drgw/sds

Model	send sendReport response sendEvent receiveEvent reportEvent
Children	receiveEvent, reportEvent, response, send, sendEvent, sendReport
Source	<pre> <xs:complexType name="interfaceSds"> <xs:annotation> <xs:documentation>DR-GW-Sds element. Use to send/receive SDS messages. Use only via SOAP.</xs:documentation> </xs:annotation> <xs:choice> <xs:element name="send" type="typeSdsSend"/> <xs:element name="sendReport" type="typeSdsSendReport"/> <xs:element name="response" type="typeResponse"/> <xs:element name="sendEvent" type="typeSdsSendEvent"/> <xs:element name="receiveEvent" type="typeSdsReceiveEvent"/> <xs:element name="reportEvent" type="typeSdsReportEvent"/> </xs:choice> </xs:complexType></pre>

Complex Type typeSdsSend

Namespace	DR-GW
Diagram	<pre> classDiagram typeRequest "0..1" -- "1..1" typeSdsSend typeSdsSend "0..1" -- "1..1" requestId typeSdsSend "0..1" -- "1..1" sds </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeSdsSend
Used by	Element interfaceSds/send
Model	requestId , sds
Children	requestId, sds
Source	<pre> <xs:complexType name="typeSdsSend"> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="sds" type="typeSds"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

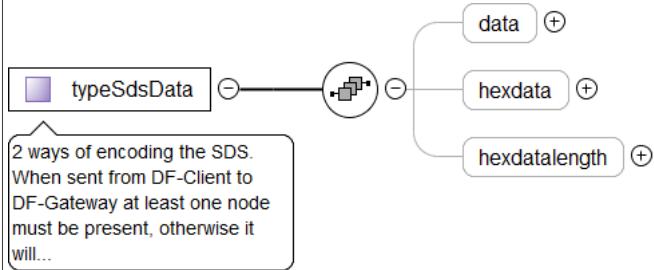
Complex Type typeSds

Namespace	DR-GW
Annotations	

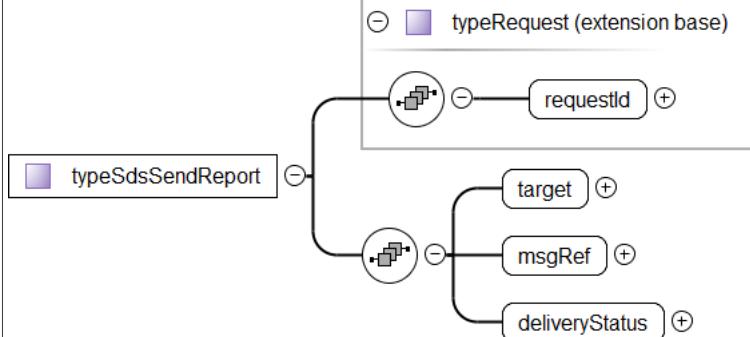
Diagram	<pre> classDiagram class typeSds { -> protocol -> sdsType -> msgRef -> report -> sdsdata -> source -> target -> forward -> validity -> tstamp -> encryption -> e2eegroup } </pre>
Used by	Elements typeSdsReceiveEvent/sds, typeSdsSend/sds, typeSdsSendEvent/sds
Model	protocol{0,1} , sdsType , msgRef{0,1} , report{0,1} , sdsdata , source{0,1} , target , forward{0,1} , validity{0,1} , tstamp{0,1} , encryption{0,1} , e2eegroup{0,1}
Children	e2eegroup, encryption, forward, msgRef, protocol, report, sdsType, sdsdata, source, target, tstamp, validity
Source	<pre> <xs:complexType name="typeSds"> <xs:annotation> <xs:documentation></xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="protocol" type="xs:unsignedByte" minOccurs="0" /> <xs:element name="sdsType" type="typeSdsType" /> <xs:element name="msgRef" type="xs:unsignedByte" minOccurs="0" /> <xs:element name="report" type="typeReport" default="none" minOccurs="0" /> <xs:element name="sdsdata" type="typeSdsData" /> <xs:element name="source" type="typeAddress" minOccurs="0" /> <xs:element name="target" type="typeAddress" /> <xs:element name="forward" type="typeAddress" minOccurs="0" /> <xs:element name="validity" type="xs:unsignedByte" minOccurs="0" /> <xs:element name="tstamp" type="xs:dateTime" minOccurs="0" /> <xs:element name="encryption" type="xs:boolean" default="true" minOccurs="0" /> <xs:element name="e2eegroup" type="typeSubscriberAddress" minOccurs="0" /> </xs:sequence> </xs:complexType> </pre>

Complex Type `typeSdsData`

Namespace	DR-GW
Annotations	<p>2 ways of encoding the SDS. When sent from DF-Client to DF-Gateway at least one node must be present, otherwise it will be discarded as not valid.</p> <p>When sent from DF-Gateway to DF-Client both nodes must be present, as it is unclear if the DF-Client supports the encoding inside raw "hexdata", so the readable decoded content must be present to.</p> <p>The default charset used within the "data" node is ISO-8859-15.</p>

Diagram	
	2 ways of encoding the SDS. When sent from DF-Client to DF-Gateway at least one node must be present, otherwise it will...
Used by	Element typeSds/sdsdata
Model	data{0,1} , hexdata{0,1} , hexdatalength{0,1}
Children	data, hexdata, hexdatalength
Source	<pre><xs:complexType name="typeSdsData"> <xs:annotation> <xs:documentation>2 ways of encoding the SDS. When sent from DF-Client to DF-Gateway at least one node must be present, otherwise it will be discarded as not valid. When sent from DF-Gateway to DF-Client both nodes must be present, as it is unclear if the DF-Client supports the encoding inside raw "hexdata", so the readable decoded content must be present to. The default charset used within the "data" node is ISO-8859-15.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="data" type="xs:string" minOccurs="0"/> <xs:element name="hexdata" type="xs:hexBinary" minOccurs="0"/> <xs:element name="hexdatalength" type="xs:integer" minOccurs="0"/> </xs:sequence> </xs:complexType></pre>

Complex Type typeSdsSendReport

Namespace	DR-GW
Diagram	
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeSdsSendReport
Used by	Element interfaceSds/sendReport
Model	requestId , target , msgRef , deliveryStatus
Children	deliveryStatus, msgRef, requestId, target
Source	<pre><xs:complexType name="typeSdsSendReport"> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="target" type="typeAddress"/> <xs:element name="msgRef" type="xs:unsignedByte"/> <xs:element name="deliveryStatus" type="xs:unsignedByte"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeSdsSendEvent

Namespace	DR-GW
-----------	-------

Annotations	An message reference is returned in the response for later message identification in case delivery and/or consume requests were asked. Is only valid for SDS-TL.
Diagram	<pre> classDiagram typeEvent < -- typeSdsSendEvent typeEvent "0..1" --> requestId typeEvent "0..1" --> result typeEvent "0..1" --> msgRef typeEvent "0..1" --> sds </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeSdsSendEvent
Used by	Element interfaceSds/sendEvent
Model	requestId{0,1} , result{0,1} , msgRef{0,1} , sds
Children	msgRef, requestId, result, sds
Source	<pre> <xss:complexType name="typeSdsSendEvent"> <xss:annotation> <xss:documentation>An message reference is returned in the response for later message identification in case delivery and/or consume requests were asked. Is only valid for SDS-TL.</xss:documentation> </xss:annotation> <xss:complexContent> <xss:extension base="typeEvent"> <xss:sequence> <xss:element name="msgRef" type="xs:unsignedByte" minOccurs="0" default="0"/> <xss:element name="sds" type="typeSds"/> </xss:sequence> </xss:extension> </xss:complexContent> </xss:complexType> </pre>

Complex Type typeSdsReceiveEvent

Namespace	DR-GW
Annotations	ReceiveEvent is fired upon received SDS.
Diagram	<pre> classDiagram typeEvent < -- typeSdsReceiveEvent typeEvent "0..1" --> requestId typeEvent "0..1" --> result typeEvent "0..1" --> sds </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeSdsReceiveEvent
Used by	Element interfaceSds/receiveEvent
Model	requestId{0,1} , result{0,1} , sds
Children	requestId, result, sds
Source	<pre> <xss:complexType name="typeSdsReceiveEvent"> <xss:annotation> <xss:documentation>ReceiveEvent is fired upon received SDS.</xss:documentation> </xss:annotation> </xss:complexType> </pre>

```

</xs:annotation>
<xs:complexContent>
  <xs:extension base="typeEvent">
    <xs:sequence>
      <xs:element name="sds" type="typeSds" />
    </xs:sequence>
  </xs:extension>
</xs:complexContent>

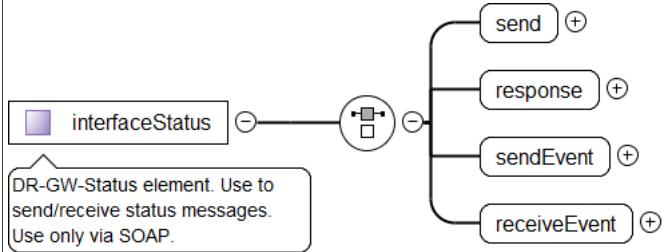
```

Complex Type typeSdsReportEvent

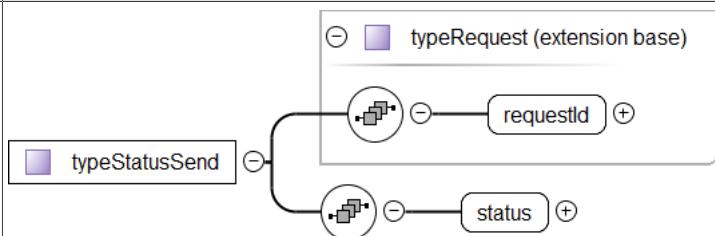
Namespace	DR-GW
Annotations	ReportEvent is fired whenever the delivery or consume report is received.
Diagram	<pre> classDiagram typeEvent < -- typeSdsReportEvent typeEvent { -requestId -result } typeSdsReportEvent { +source +target +msgRef +deliveryStatus +tstamp } note over typeSdsReportEvent: ReportEvent is fired whenever the delivery or consume report is received. </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeSdsReportEvent
Used by	Element interfaceSds/reportEvent
Model	requestId{0,1} , result{0,1} , source , target , msgRef , deliveryStatus , tstamp
Children	deliveryStatus, msgRef, requestId, result, source, target, tstamp
Source	<pre> <xs:complexType name="typeSdsReportEvent"> <xs:annotation> <xs:documentation>ReportEvent is fired whenever the delivery or consume report is received.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="source" type="typeAddress"/> <xs:element name="target" type="typeAddress"/> <xs:element name="msgRef" type="xs:unsignedByte"/> <xs:element name="deliveryStatus" type="xs:unsignedByte"/> <xs:element name="tstamp" type="xs:dateTime"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type interfaceStatus

Namespace	DR-GW
Annotations	DR-GW-Status element. Use to send/receive status messages. Use only via SOAP.

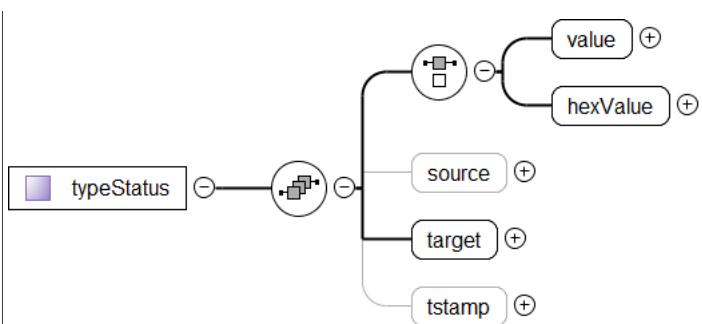
Diagram	
Used by	Element drgw/status
Model	send response sendEvent receiveEvent
Children	receiveEvent, response, send, sendEvent
Source	<pre><xs:complexType name="interfaceStatus"> <xs:annotation> <xs:documentation>DR-GW-Status element. Use to send/receive status messages. Use only via SOAP.</xs:documentation> </xs:annotation> <xs:choice> <xs:element name="send" type="typeStatusSend"/> <xs:element name="response" type="typeResponse"/> <xs:element name="sendEvent" type="typeStatusSendEvent"/> <xs:element name="receiveEvent" type="typeStatusReceiveEvent"/> </xs:choice> </xs:complexType></pre>

Complex Type typeStatusSend

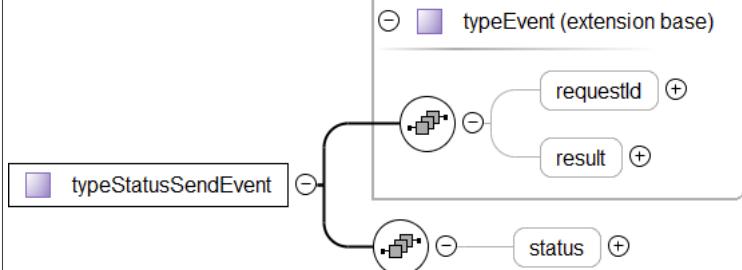
Namespace	DR-GW
Diagram	
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeStatusSend
Used by	Element interfaceStatus/send
Model	requestId, status
Children	requestId, status
Source	<pre><xs:complexType name="typeStatusSend"> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="status" type="typeStatus"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeStatus

Namespace	DR-GW
-----------	-------

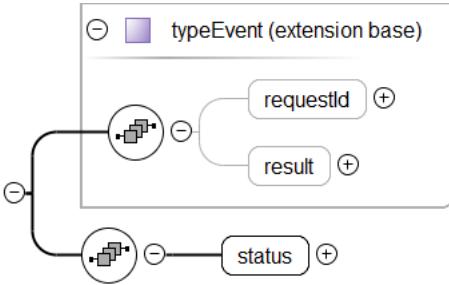
Diagram	
Used by	Elements typeStatusReceiveEvent/status, typeStatusSendEvent/status, typeStatusSendEventEvent/status
Model	(value hexValue) , source{0,1} , target , tstamp{0,1}
Children	hexValue, source, target, tstamp, value
Source	<pre><xs:complexType name="typeStatus"> <xs:sequence> <xs:choice> <xs:element name="value" type="xs:unsignedShort"/> <xs:element name="hexValue" type="xs:hexBinary"/> </xs:choice> <xs:element name="source" type="typeAddress" minOccurs="0"/> <xs:element name="target" type="typeAddress"/> <xs:element name="tstamp" type="xs:dateTime" minOccurs="0"/> </xs:sequence> </xs:complexType></pre>

Complex Type typeStatusSendEvent

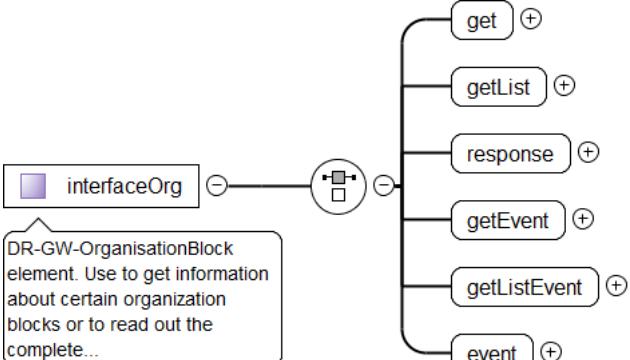
Namespace	DR-GW
Diagram	
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeStatusSendEvent
Used by	Element interfaceStatus/sendEvent
Model	requestId{0,1} , result{0,1} , status{0,1}
Children	requestId, result, status
Source	<pre><xs:complexType name="typeStatusSendEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="status" type="typeStatus" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeStatusReceiveEvent

Namespace	DR-GW
-----------	-------

Diagram	
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeStatusReceiveEvent
Used by	Element interfaceStatus/receiveEvent
Model	requestId{0,1} , result{0,1} , status
Children	requestId, result, status
Source	<pre><xs:complexType name="typeStatusReceiveEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="status" type="typeStatus"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type interfaceOrg

Namespace	DR-GW
Annotations	DR-GW-OrganisationBlock element. Use to get information about certain organization blocks or to read out the complete organization schema the DF-Client has rights for. Use only via SOAP.
Diagram	 <p>DR-GW-OrganisationBlock element. Use to get information about certain organization blocks or to read out the complete...</p>
Used by	Element drgw/org
Model	get getList response getEvent getListEvent event
Children	event, get, getEvent, getList, getListEvent, response
Source	<pre><xs:complexType name="interfaceOrg"> <xs:annotation> <xs:documentation>DR-GW-OrganisationBlock element. Use to get information about certain organization blocks or to read out the complete organization schema the DF-Client has rights for. Use only via SOAP.</xs:documentation> </xs:annotation> <xs:choice> <xs:element name="get" type="typeOrgGet"/> <xs:element name="getList" type="typeOrgGetList"/> <xs:element name="response" type="typeResponse"/> <xs:element name="getEvent" type="typeOrgGetEvent"/> <xs:element name="getListEvent" type="typeOrgGetListEvent"/> <xs:element name="event" type="typeOrgEvent"/> </xs:choice> </xs:complexType></pre>

Complex Type typeOrgGet

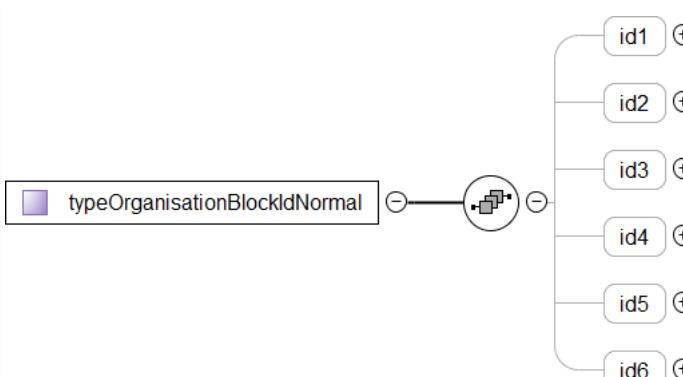
Namespace	DR-GW
Diagram	<pre> classDiagram typeRequest < -- typeOrgGet typeOrgGet { -requestId -orgblockId } </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeOrgGet
Used by	Element interfaceOrg/get
Model	requestId , orgblockId
Children	orgblockId, requestId
Source	<pre> <xs:complexType name="typeOrgGet"> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="orgblockId" type="typeOrganisationBlockId"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeOrganisationBlockId

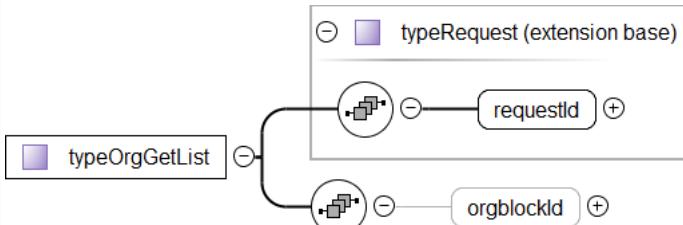
Namespace	DR-GW
Annotations	
Diagram	<pre> typeOrganisationBlockId { --orgblockId --orgblockIdSimple } </pre>
Used by	Elements typeAppGetList/orgblockId, typeApplication/orgblockId, typeGroup/orgblockId, typeGroupGetList/orgblockId, typeOrgGet/orgblockId, typeOrgGetList/orgblockId, typeOrganisationBlock/orgblockId, typeRadio/orgblockId, typeRadioGetList/orgblockId
Model	orgblockId orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Source	<pre> <xs:complexType name="typeOrganisationBlockId"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:choice> <xs:element name="orgblockId" type="typeOrganisationBlockIdNormal"/> <xs:element name="orgblockIdSimple" type="typeOrganisationBlockIdSimple"/> </xs:choice> </xs:complexType> </pre>

Complex Type typeOrganisationBlockIdNormal

Namespace	DR-GW
Annotations	

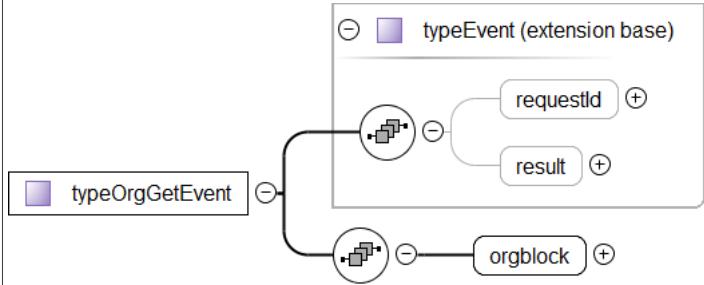
Diagram	
Used by	Element typeOrganisationBlockId/orgblockId
Model	id1{0,1} , id2{0,1} , id3{0,1} , id4{0,1} , id5{0,1} , id6{0,1}
Children	id1, id2, id3, id4, id5, id6
Source	<pre><xs:complexType name="typeOrganisationBlockIdNormal"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="id1" type="xs:unsignedShort" minOccurs="0"/> <xs:element name="id2" type="xs:unsignedShort" minOccurs="0"/> <xs:element name="id3" type="xs:unsignedShort" minOccurs="0"/> <xs:element name="id4" type="xs:unsignedShort" minOccurs="0"/> <xs:element name="id5" type="xs:unsignedShort" minOccurs="0"/> <xs:element name="id6" type="xs:unsignedShort" minOccurs="0"/> </xs:sequence> </xs:complexType></pre>

Complex Type typeOrgGetList

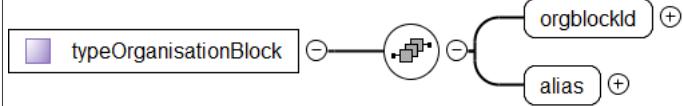
Namespace	DR-GW
Diagram	
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeOrgGetList
Used by	Element interfaceOrg/getList
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId
Source	<pre><xs:complexType name="typeOrgGetList"> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeOrgGetEvent

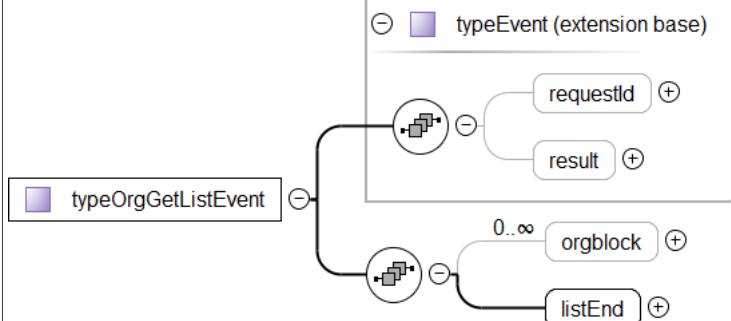
Namespace	DR-GW
-----------	-------

Diagram	
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeOrgGetEvent
Used by	Element interfaceOrg/getEvent
Model	requestId{0,1} , result{0,1} , orgblock
Children	orgblock, requestId, result
Source	<pre><xs:complexType name="typeOrgGetEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="orgblock" type="typeOrganisationBlock"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeOrganisationBlock

Namespace	DR-GW
Annotations	
Diagram	
Used by	Elements typeOrgEvent/orgblock, typeOrgGetEvent/orgblock, typeOrgGetListEvent/orgblock
Model	orgblockId , alias
Children	alias, orgblockId
Source	<pre><xs:complexType name="typeOrganisationBlock"> <xs:annotation> <xs:documentation></xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="orgblockId" type="typeOrganisationBlockId"/> <xs:element name="alias" type="xs:normalizedString"/> </xs:sequence> </xs:complexType></pre>

Complex Type typeOrgGetListEvent

Namespace	DR-GW
Diagram	

Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeOrgGetListEvent
Used by	Element interfaceOrg/getListEvent
Model	requestId{0,1} , result{0,1} , orgblock* , listEnd
Children	listEnd, orgblock, requestId, result
Source	<pre><xs:complexType name="typeOrgGetListEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="orgblock" type="typeOrganisationBlock" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="listEnd" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

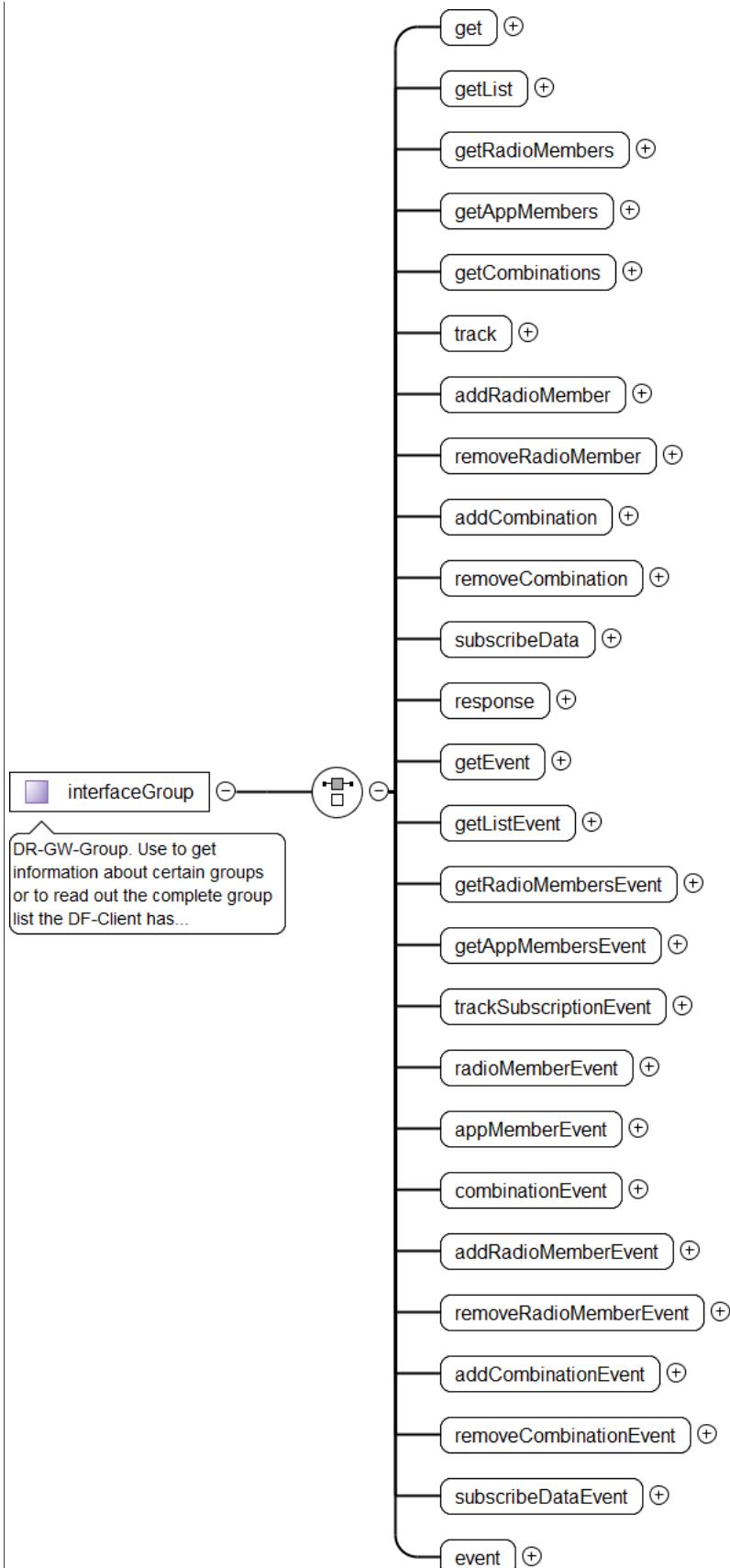
Complex Type typeOrgEvent

Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeOrgEvent typeEvent "0..1" -- "1..1" requestId typeEvent "0..1" -- "1..1" result typeOrgEvent "0..1" -- "1..1" orgblock typeOrgEvent "0..1" -- "1..1" delete </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeOrgEvent
Used by	Element interfaceOrg/event
Model	requestId{0,1} , result{0,1} , orgblock , delete
Children	delete, orgblock, requestId, result
Source	<pre><xs:complexType name="typeOrgEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="orgblock" type="typeOrganisationBlock"/> <xs:element name="delete" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type interfaceGroup

Namespace	DR-GW
Annotations	DR-GW-Group. Use to get information about certain groups or to read out the complete group list the DF-Client has rights for. There are also many other methods to execute over a certain group. See each method description. Use only via SOAP.

Diagram



Used by	Element	drgw/group
Model	get getList getRadioMembers getAppMembers getCombinations track addRadioMember removeRadioMember addCombination removeCombination subscribeData response getEvent getListEvent getRadioMembersEvent getAppMembersEvent	

	trackSubscriptionEvent radioMemberEvent appMemberEvent combinationEvent addRadioMemberEvent removeRadioMemberEvent addCombinationEvent removeCombinationEvent subscribeDataEvent event
Children	addCombination, addCombinationEvent, addRadioMember, addRadioMemberEvent, appMemberEvent, combinationEvent, event, get, getAppMembers, getAppMembersEvent, getCombinations, getEvent, getList, getListEvent, getRadioMembers, getRadioMembersEvent, radioMemberEvent, removeCombination, removeCombinationEvent, removeRadioMember, removeRadioMemberEvent, response, subscribeData, subscribeDataEvent, track, trackSubscriptionEvent
Source	<pre> <xs:complexType name="interfaceGroup"> <xs:annotation> <xs:documentation>DR-GW-Group. Use to get information about certain groups or to read out the complete group list the DF-Client has rights for. There are also many other methods to execute over a certain group. See each method description. Use only via SOAP.</xs:documentation> </xs:annotation> <xs:choice> <xs:element name="get" type="typeGroupGet"/> <xs:element name="getList" type="typeGroupGetList"/> <xs:element name="getRadioMembers" type="typeGroupGetRadioMembers"/> <xs:element name="getAppMembers" type="typeGroupGetAppMembers"/> <xs:element name="getCombinations" type="typeGroupGetCombinations"/> <xs:element name="track" type="typeGroupTrack"/> <xs:element name="addRadioMember" type="typeGroupAddRadioMember"/> <xs:element name="removeRadioMember" type="typeGroupRemoveRadioMember"/> <xs:element name="addCombination" type="typeGroupAddCombination"/> <xs:element name="removeCombination" type="typeGroupRemoveCombination"/> <xs:element name="subscribeData" type="typeGroupSubscribeData"/> <xs:element name="response" type="typeResponse"/> <xs:element name="getEvent" type="typeGroupGetEvent"/> <xs:element name="getListEvent" type="typeGroupGetListEvent"/> <xs:element name="getRadioMembersEvent" type="typeGroupGetRadioMembersEvent"/> <xs:element name="getAppMembersEvent" type="typeGroupGetAppMembersEvent"/> <xs:element name="trackSubscriptionEvent" type="typeGroupTrackSubscriptionEvent"/> <xs:element name="radioMemberEvent" type="typeGroupRadioMemberEvent"/> <xs:element name="appMemberEvent" type="typeGroupAppMemberEvent"/> <xs:element name="combinationEvent" type="typeGroupCombinationEvent"/> <xs:element name="addRadioMemberEvent" type="typeGroupAddRadioMemberEvent"/> <xs:element name="removeRadioMemberEvent" type="typeGroupRemoveRadioMemberEvent"/> <xs:element name="addCombinationEvent" type="typeGroupAddCombinationEvent"/> <xs:element name="removeCombinationEvent" type="typeGroupRemoveCombinationEvent"/> <xs:element name="subscribeDataEvent" type="typeGroupSubscribeDataEvent"/> <xs:element name="event" type="typeGroupEvent"/> </xs:choice> </xs:complexType></pre>

Complex Type typeGroupGet

Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeRequest "0..1" -- "1..1" typeGroupGet typeGroupGet "0..1" -- "1..1" requestId typeGroupGet "0..1" -- "1..1" group </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupGet
Used by	Element interfaceGroup/get
Model	requestId , group
Children	group, requestId
Source	<pre> <xs:complexType name="typeGroupGet"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

```

</xs:complexContent>
</xs:complexType>

```

Complex Type typeGroupGetList

Namespace	DR-GW
Annotations	
Diagram	<pre> sequenceDiagram participant DRGW as DR-GW DRGW->>DRGW: typeGroupGetList activate DRGW DRGW->>DRGW: typeRequest (extension base) deactivate DRGW DRGW-->>DRGW: requestId DRGW-->>DRGW: orgblockId </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupGetList
Used by	Element interfaceGroup/getList
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId
Source	<pre> <xs:complexType name="typeGroupGetList"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeGroupGetRadioMembers

Namespace	DR-GW
Annotations	
Diagram	<pre> sequenceDiagram participant DRGW as DR-GW DRGW->>DRGW: typeGroupGetRadioMembers activate DRGW DRGW->>DRGW: typeRequest (extension base) deactivate DRGW DRGW-->>DRGW: requestId DRGW-->>DRGW: group </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupGetRadioMembers
Used by	Element interfaceGroup/getRadioMembers
Model	requestId , group
Children	group, requestId
Source	<pre> <xs:complexType name="typeGroupGetRadioMembers"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress" /> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

```

</xs:extension>
</xs:complexContent>
</xs:complexType>

```

Complex Type typeGroupGetAppMembers

Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeRequest < -- typeGroupGetAppMembers typeRequest "0..1" --> "1..1" requestId typeRequest "0..1" --> "1..1" group </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupGetAppMembers
Used by	Element interfaceGroup/getAppMembers
Model	requestId , group
Children	group, requestId
Source	<pre> <xs:complexType name="typeGroupGetAppMembers"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress" /> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeGroupGetCombinations

Namespace	DR-GW
Annotations	The method requests the groups that belong to the same combined group as the group specified.
Diagram	<p>The method requests the groups that belong to the same combined group as the group specified.</p> <pre> classDiagram typeRequest < -- typeGroupGetCombinations typeRequest "0..1" --> "1..1" requestId typeRequest "0..1" --> "1..1" group </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupGetCombinations
Used by	Element interfaceGroup/getCombinations
Model	requestId , group
Children	group, requestId
Source	<pre> <xs:complexType name="typeGroupGetCombinations"> <xs:annotation> <xs:documentation>The method requests the groups that belong to the same combined group as the group specified.</xs:documentation> </xs:annotation> </xs:complexType> </pre>

```

<xs:complexContent>
  <xs:extension base="typeRequest">
    <xs:sequence>
      <xs:element name="group" type="typeSubscriberAddress"/>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>

```

Complex Type typeGroupTrack

Namespace	DR-GW
Annotations	
Diagram	<pre> graph TD typeRequest["typeRequest (extension base)"] typeGroupTrack["typeGroupTrack"] requestId["requestId"] group["group"] mask["mask"] stop["stop"] typeGroupTrack --> typeRequest typeGroupTrack -.- requestId typeGroupTrack -.- group typeGroupTrack -.- mask typeGroupTrack -.- stop </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupTrack
Used by	Element interfaceGroup/track
Model	requestId , group , mask , stop
Children	group, mask, requestId, stop
Source	<pre> <xs:complexType name="typeGroupTrack"> <xs:annotation> <xs:documentation> Requests the addition of a radio subscriber to a group. This might cause DGNA operation in the air interface. </xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> <xs:element name="mask" type="typeGroupTrackingMask"/> <xs:element name="stop" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeGroupAddRadioMember

Namespace	DR-GW
Annotations	<p>Requests the addition of a radio subscriber to a group. This might cause DGNA operation in the air interface.</p>
Diagram	<pre> graph TD typeRequest["typeRequest (extension base)"] typeGroupAddRadioMember["typeGroupAddRadioMember"] requestId["requestId"] radio["radio"] group["group"] membership["membership"] typeGroupAddRadioMember --> typeRequest typeGroupAddRadioMember -.- requestId typeGroupAddRadioMember -.- radio typeGroupAddRadioMember -.- group typeGroupAddRadioMember -.- membership </pre> <p>Requests the addition of a radio subscriber to a group. This might cause DGNA operation in the air interface.</p>

Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupAddRadioMember
Used by	Element interfaceGroup/addRadioMember
Model	requestId , radio , group , membership{0,1}
Children	group, membership, radio, requestId
Source	<pre><xs:complexType name="typeGroupAddRadioMember"> <xs:annotation> <xs:documentation>Requests the addition of a radio subscriber to a group. This might cause DGNA operation in the air interface.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="group" type="typeSubscriberAddress"/> <xs:element name="membership" type="typeMembershipType" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeGroupRemoveRadioMember

Namespace	DR-GW
Annotations	Requests removing a radio subscriber from a group. This might cause DGNA operation in the air interface.
Diagram	<pre> classDiagram typeRequest --> typeGroupRemoveRadioMember typeGroupRemoveRadioMember "Requests removing a radio subscriber from a group. This might cause DGNA operation in the air interface." typeGroupRemoveRadioMember <> requestId typeGroupRemoveRadioMember <> radio typeGroupRemoveRadioMember <> group </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupRemoveRadioMember
Used by	Element interfaceGroup/removeRadioMember
Model	requestId , radio , group
Children	group, radio, requestId
Source	<pre><xs:complexType name="typeGroupRemoveRadioMember"> <xs:annotation> <xs:documentation>Requests removing a radio subscriber from a group. This might cause DGNA operation in the air interface.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="group" type="typeSubscriberAddress"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeGroupAddCombination

Namespace	DR-GW
Annotations	Requests the addition of a group to a combined group.

Diagram	<pre> classDiagram typeRequest < -- typeGroupAddCombination typeGroupAddCombination { requestId group baseGroup force } </pre> <p>Requests the addition of a group to a combined group.</p>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupAddCombination
Used by	Element interfaceGroup/addCombination
Model	requestId , group , baseGroup , force{0,1}
Children	baseGroup, force, group, requestId
Source	<pre> <xss:complexType name="typeGroupAddCombination"> <xss:annotation> <xss:documentation>Requests the addition of a group to a combined group.</xss:documentation> </xss:annotation> <xss:complexContent> <xss:extension base="typeRequest"> <xss:sequence> <xss:element name="group" type="typeSubscriberAddress"/> <xss:element name="baseGroup" type="typeSubscriberAddress"/> <xss:element name="force" type="xs:boolean" minOccurs="0" default="true"/> </xss:sequence> </xss:extension> </xss:complexContent> </xss:complexType> </pre>

Complex Type typeGroupRemoveCombination

Namespace	DR-GW
Annotations	Requests removing a group from a combined group.
Diagram	<pre> classDiagram typeRequest < -- typeGroupRemoveCombination typeGroupRemoveCombination { requestId group baseGroup } </pre> <p>Requests removing a group from a combined group.</p>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupRemoveCombination
Used by	Element interfaceGroup/removeCombination
Model	requestId , group , baseGroup
Children	baseGroup, group, requestId
Source	<pre> <xss:complexType name="typeGroupRemoveCombination"> <xss:annotation> <xss:documentation>Requests removing a group from a combined group.</xss:documentation> </xss:annotation> <xss:complexContent> <xss:extension base="typeRequest"> <xss:sequence> <xss:element name="group" type="typeSubscriberAddress"/> <xss:element name="baseGroup" type="typeSubscriberAddress"/> </xss:sequence> </xss:extension> </xss:complexContent> </xss:complexType> </pre>

```

</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

Complex Type typeGroupSubscribeData

Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeRequest < -- typeGroupSubscribeData typeRequest "1..1" --> requestId typeGroupSubscribeData "1..infinity" --> group </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeGroupSubscribeData
Used by	Element interfaceGroup/subscribeData
Model	requestId , group+
Children	group, requestId
Source	<pre> <xs:complexType name="typeGroupSubscribeData"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="group" type="typeGroupDataSubscription" maxOccurs="unbounded" /> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeGroupDataSubscription

Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeGroupDataSubscription "1..1" --> addr typeGroupDataSubscription "1..1" --> useSDS typeGroupDataSubscription "1..1" --> useStatus </pre>
Used by	Elements typeGroupSubscribeData/group, typeGroupSubscribeDataEvent/group
Model	addr , useSDS , useStatus
Children	addr, useSDS, useStatus
Source	<pre> <xs:complexType name="typeGroupDataSubscription"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="addr" type="typeSubscriberAddress" /> <xs:element name="useSDS" type="xs:boolean" /> <xs:element name="useStatus" type="xs:boolean" /> </xs:sequence> </xs:complexType> </pre>

Complex Type typeGroupGetEvent

Namespace	DR-GW
-----------	-------

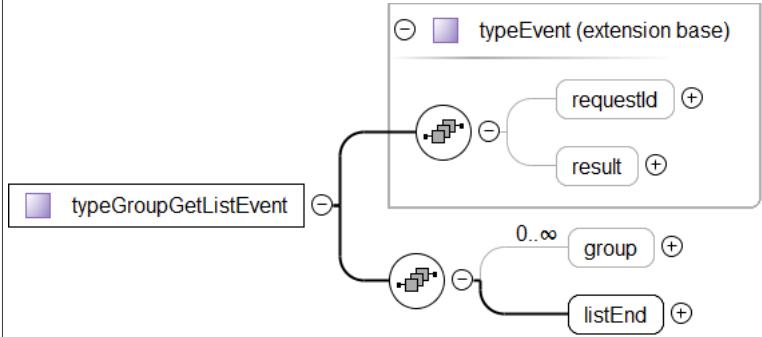
Annotations	
Diagram	<pre> classDiagram typeEvent { requestId result } typeGroupGetEvent { group } typeEvent < -- typeGroupGetEvent </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupGetEvent
Used by	Element interfaceGroup/getEvent
Model	requestId{0,1} , result{0,1} , group
Children	group, requestId, result
Source	<pre> <xss:complexType name="typeGroupGetEvent"> <xss:annotation> <xss:documentation/> </xss:annotation> <xss:complexContent> <xss:extension base="typeEvent"> <xss:sequence> <xss:element name="group" type="typeGroup" /> </xss:sequence> </xss:extension> </xss:complexContent> </xss:complexType> </pre>

Complex Type typeGroup

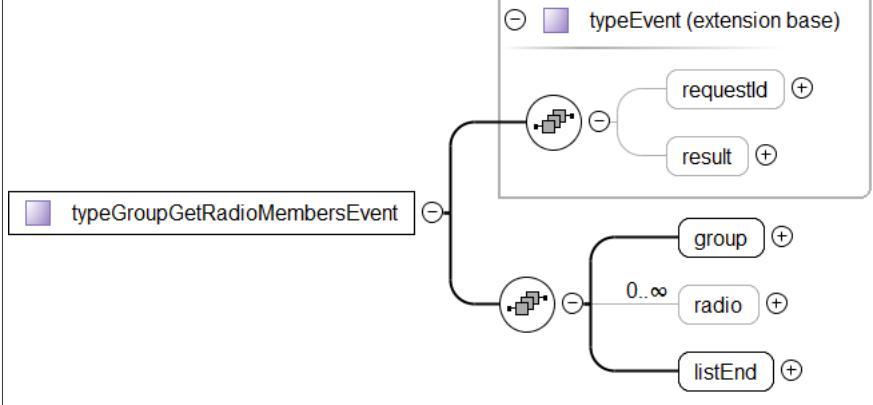
Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeGroup { addr alias orgblockId } typeGroup <--> addr typeGroup <--> alias typeGroup <--> orgblockId </pre>
Used by	Elements typeGroupAppMemberEvent/group, typeGroupEvent/group, typeGroupGetEvent/group, typeGroupGetListEvent/group
Model	addr , alias , orgblockId
Children	addr, alias, orgblockId
Source	<pre> <xss:complexType name="typeGroup"> <xss:annotation> <xss:documentation/> </xss:annotation> <xss:sequence> <xss:element name="addr" type="typeSubscriberAddress" /> <xss:element name="alias" type="xs:normalizedString" /> <xss:element name="orgblockId" type="typeOrganisationBlockId" /> </xss:sequence> </xss:complexType> </pre>

Complex Type typeGroupGetListEvent

Namespace	DR-GW
Annotations	

Diagram	
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupGetListEvent
Used by	Element interfaceGroup/getListEvent
Model	requestId{0,1} , result{0,1} , group* , listEnd
Children	group, listEnd, requestId, result
Source	<pre><xs:complexType name="typeGroupGetListEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeGroup" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="listEnd" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeGroupGetRadioMembersEvent

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupGetRadioMembersEvent
Used by	Element interfaceGroup/getRadioMembersEvent
Model	requestId{0,1} , result{0,1} , group , radio* , listEnd
Children	group, listEnd, radio, requestId, result
Source	<pre><xs:complexType name="typeGroupGetRadioMembersEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeRadio" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="listEnd" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

```
<xs:extension base="typeEvent">
  <xs:sequence>
    <xs:element name="group" type="typeSubscriberAddress"/>
    <xs:element name="radio" type="typeSubscriberAddress" minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="listEnd" type="xs:boolean"/>
  </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

Complex Type typeGroupGetAppMembersEvent

Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeEvent < -- typeGroupGetAppMembersEvent typeEvent { -requestId -result +app +listEnd } typeGroupGetAppMembersEvent { +app } </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent <ul style="list-style-type: none"> • typeGroupGetAppMembersEvent
Used by	Element interfaceGroup/getAppMembersEvent
Model	requestId{0,1} , result{0,1} , app* , listEnd
Children	app, listEnd, requestId, result
Source	<pre> <xs:complexType name="typeGroupGetAppMembersEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="app" type="typeSubscriberAddress" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="listEnd" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeGroupTrackSubscriptionEvent

Namespace	DR-GW
Annotations	
Diagram	<p>The diagram illustrates two UML class definitions:</p> <ul style="list-style-type: none"> typeEvent (extension base): This class has two associations. One association, marked with a minus sign (-), connects to a class with a plus sign (+) icon, labeled requestId. Another association, also marked with a minus sign (-), connects to a class with a plus sign (+) icon, labeled result. typeGroupTrackSubscriptionEvent: This class has three associations. One association, marked with a minus sign (-), connects to a class with a plus sign (+) icon, labeled group. Another association, marked with a minus sign (-), connects to a class with a plus sign (+) icon, labeled mask. A third association, marked with a minus sign (-), connects to a class with a plus sign (+) icon, labeled stop.

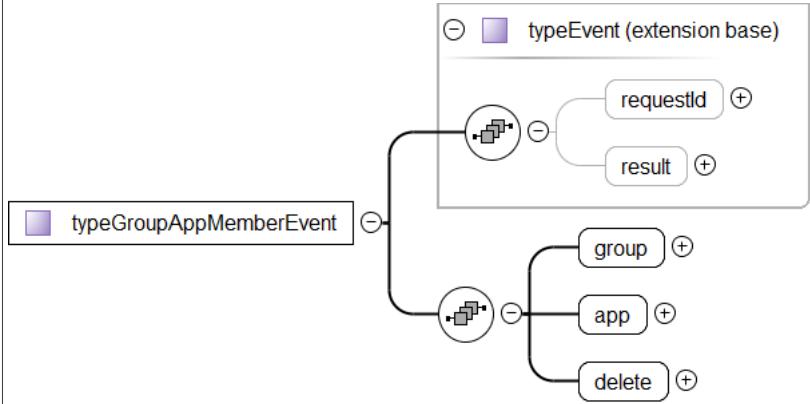
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupTrackSubscriptionEvent
Used by	Element interfaceGroup/trackSubscriptionEvent
Model	requestId{0,1} , result{0,1} , group , mask , stop
Children	group, mask, requestId, result, stop
Source	<pre><xs:complexType name="typeGroupTrackSubscriptionEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> <xs:element name="mask" type="typeGroupTrackingMask"/> <xs:element name="stop" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeGroupRadioMemberEvent

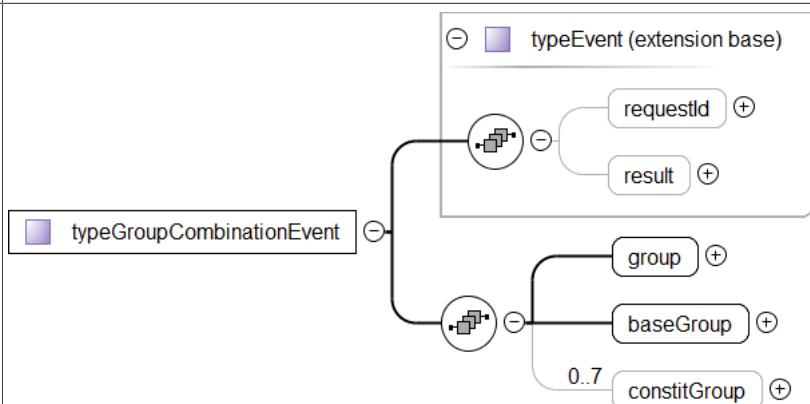
Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeEvent < -- typeGroupRadioMemberEvent typeEvent "0..1" -- "0..1" typeGroupRadioMemberEvent : requestId typeEvent "0..1" -- "0..1" typeGroupRadioMemberEvent : result typeEvent "0..1" -- "0..1" typeGroupRadioMemberEvent : group typeEvent "0..1" -- "0..1" typeGroupRadioMemberEvent : radio typeEvent "0..1" -- "0..1" typeGroupRadioMemberEvent : delete </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupRadioMemberEvent
Used by	Element interfaceGroup/radioMemberEvent
Model	requestId{0,1} , result{0,1} , group , radio , delete
Children	delete, group, radio, requestId, result
Source	<pre><xs:complexType name="typeGroupRadioMemberEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="delete" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeGroupAppMemberEvent

Namespace	DR-GW
Annotations	

Diagram	
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupAppMemberEvent
Used by	Element interfaceGroup/appMemberEvent
Model	requestId{0,1} , result{0,1} , group , app , delete
Children	app, delete, group, requestId, result
Source	<pre><xs:complexType name="typeGroupAppMemberEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeGroup"/> <xs:element name="app" type="typeSubscriberAddress"/> <xs:element name="delete" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeGroupCombinationEvent

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupCombinationEvent
Used by	Element interfaceGroup/combinationEvent
Model	requestId{0,1} , result{0,1} , group , baseGroup , constitGroup{0,7}
Children	baseGroup, constitGroup, group, requestId, result
Source	<pre><xs:complexType name="typeGroupCombinationEvent"></pre>

```

<xs:annotation>
  <xs:documentation/>
</xs:annotation>
<xs:complexType>
  <xs:extension base="typeEvent">
    <xs:sequence>
      <xs:element name="group" type="typeSubscriberAddress" />
      <xs:element name="baseGroup" type="typeSubscriberAddress" />
      <xs:element name="constitGroup" type="typeSubscriberAddress" minOccurs="0" maxOccurs="7" />
    </xs:sequence>
  </xs:extension>
</xs:complexType>
</xs:complexType>

```

Complex Type typeGroupAddRadioMemberEvent

Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeEvent --> typeGroupAddRadioMemberEvent typeGroupAddRadioMemberEvent { requestId result radio group } </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupAddRadioMemberEvent
Used by	Element interfaceGroup/addRadioMemberEvent
Model	requestId{0,1} , result{0,1} , radio , group
Children	group, radio, requestId, result
Source	<pre> <xs:complexType name="typeGroupAddRadioMemberEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress" /> <xs:element name="group" type="typeSubscriberAddress" /> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeGroupRemoveRadioMemberEvent

Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeEvent --> typeGroupRemoveRadioMemberEvent typeGroupRemoveRadioMemberEvent { requestId result radio group } </pre>
Source	

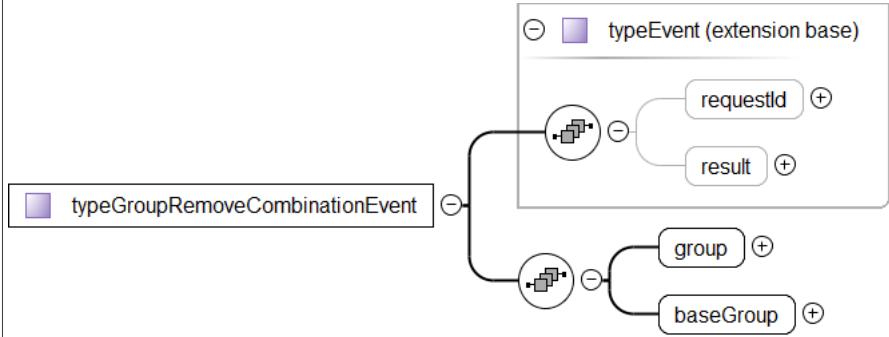
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupRemoveRadioMemberEvent
Used by	Element interfaceGroup/removeRadioMemberEvent
Model	requestId{0,1} , result{0,1} , radio , group
Children	group, radio, requestId, result
Source	<pre><xs:complexType name="typeGroupRemoveRadioMemberEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress" /> <xs:element name="group" type="typeSubscriberAddress" /> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeGroupAddCombinationEvent

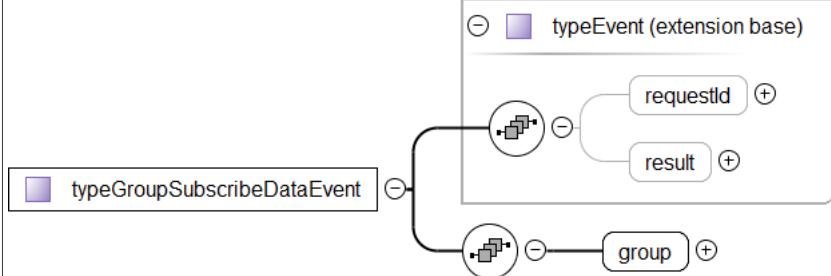
Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeEvent --> typeGroupAddCombinationEvent typeGroupAddCombinationEvent < -- typeEvent typeGroupAddCombinationEvent < -- typeEvent typeGroupAddCombinationEvent < -- typeEvent typeGroupAddCombinationEvent < -- typeEvent </pre> <p>The diagram illustrates the inheritance relationship between the typeEvent extension base and the typeGroupAddCombinationEvent type. The typeGroupAddCombinationEvent type is shown as an extension of typeEvent, indicated by a line with a hollow arrowhead pointing from typeEvent to typeGroupAddCombinationEvent. The typeGroupAddCombinationEvent type itself is enclosed in a box with a solid border. Inside this box, there are four attributes represented by rounded rectangles with plus signs: requestId, result, group, and baseGroup.</p>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupAddCombinationEvent
Used by	Element interfaceGroup/addCombinationEvent
Model	requestId{0,1} , result{0,1} , group , baseGroup
Children	baseGroup, group, requestId, result
Source	<pre><xs:complexType name="typeGroupAddCombinationEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress" /> <xs:element name="baseGroup" type="typeSubscriberAddress" /> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeGroupRemoveCombinationEvent

Namespace	DR-GW
Annotations	

Diagram	
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupRemoveCombinationEvent
Used by	Element interfaceGroup/removeCombinationEvent
Model	requestId{0,1} , result{0,1} , group , baseGroup
Children	baseGroup, group, requestId, result
Source	<pre><xs:complexType name="typeGroupRemoveCombinationEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> <xs:element name="baseGroup" type="typeSubscriberAddress"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeGroupSubscribeDataEvent

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupSubscribeDataEvent
Used by	Element interfaceGroup/subscribeDataEvent
Model	requestId{0,1} , result{0,1} , group
Children	group, requestId, result
Source	<pre><xs:complexType name="typeGroupSubscribeDataEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeGroupDataSubscription"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

```

</xs:complexContent>
</xs:complexType>

```

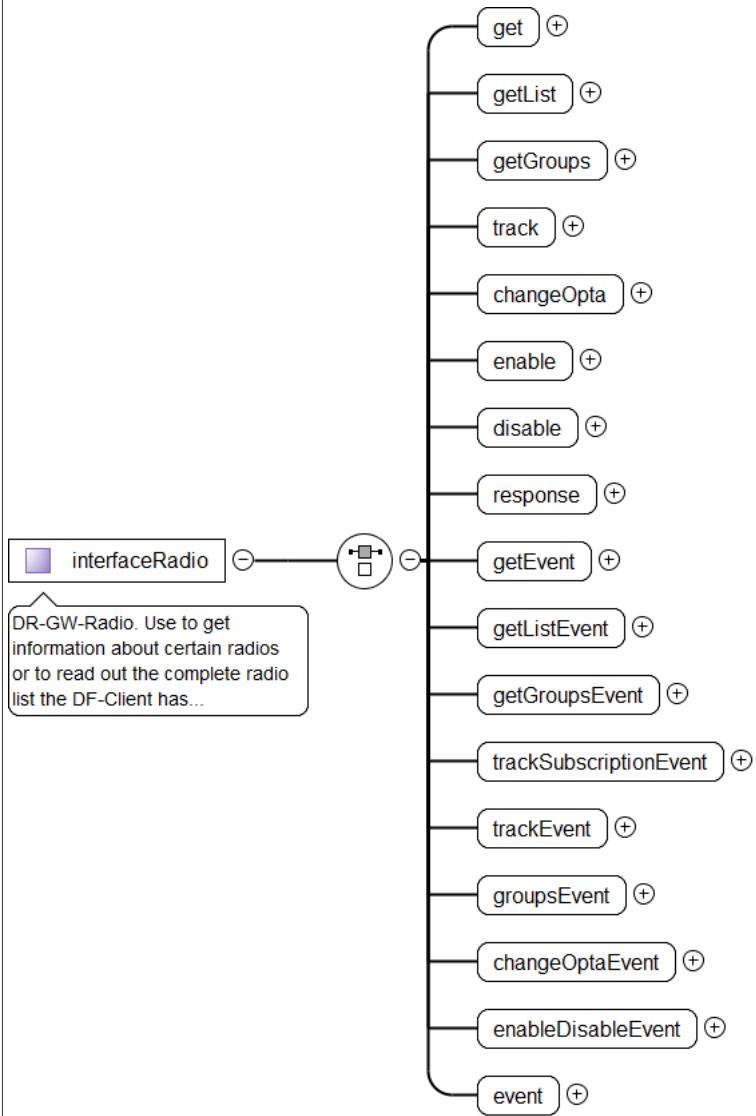
Complex Type typeGroupEvent

Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeEvent < -- typeGroupEvent typeEvent "0..1" *--> requestId typeEvent "0..1" *--> result typeGroupEvent "0..1" *--> group typeGroupEvent "0..1" *--> delete </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupEvent
Used by	Element interfaceGroup/event
Model	requestId{0,1} , result{0,1} , group , delete
Children	delete, group, requestId, result
Source	<pre> <xs:complexType name="typeGroupEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeGroup"/> <xs:element name="delete" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type interfaceRadio

Namespace	DR-GW
Annotations	DR-GW-Radio. Use to get information about certain radios or to read out the complete radio list the DF-Client has rights for. There are also many other methods to execute over a certain radio. See each method description. Use only via SOAP.

Diagram



Used by	Element drgw/radio
Model	get getList getGroups track changeOpta enable disable response getEvent getListEvent getGroupsEvent trackSubscriptionEvent trackEvent groupsEvent changeOptaEvent enableDisableEvent event
Children	changeOpta, changeOptaEvent, disable, enable, enableDisableEvent, event, get, getEvent, getGroups, getGroupsEvent, getList, getListEvent, groupsEvent, response, track, trackEvent, trackSubscriptionEvent
Source	<pre> <xs:complexType name="interfaceRadio"> <xs:annotation> <xs:documentation>DR-GW-Radio. Use to get information about certain radios or to read out the complete radio list the DF-Client has rights for. There are also many other methods to execute over a certain radio. See each method description. Use only via SOAP.</xs:documentation> <xs:annotation> <xs:choice> <xs:element name="get" type="typeRadioGet"/> <xs:element name="getList" type="typeRadioGetList"/> <xs:element name="getGroups" type="typeRadioGetGroups"/> <xs:element name="track" type="typeRadioTrack"/> <xs:element name="changeOpta" type="typeRadioChangeOpta"/> <xs:element name="enable" type="typeRadioEnable"/> <xs:element name="disable" type="typeRadioDisable"/> <xs:element name="response" type="typeResponse"/> <xs:element name="getEvent" type="typeRadioGetEvent"/> <xs:element name="getListEvent" type="typeRadioGetListEvent"/> <xs:element name="getGroupsEvent" type="typeRadioGetGroupsEvent"/> <xs:element name="trackSubscriptionEvent" type="typeRadioTrackSubscriptionEvent"/> <xs:element name="trackEvent" type="typeRadioTrackEvent"/> <xs:element name="groupsEvent" type="typeRadioGroupsEvent"/> <xs:element name="changeOptaEvent" type="typeRadioChangeOptaEvent"/> <xs:element name="enableDisableEvent" type="typeRadioEnableDisableEvent"/> <xs:element name="event" type="typeRadioEvent"/> </xs:choice> </xs:annotation> </xs:complexType> </pre>

```

</xs:choice>
</xs:complexType>

```

Complex Type typeRadioGet

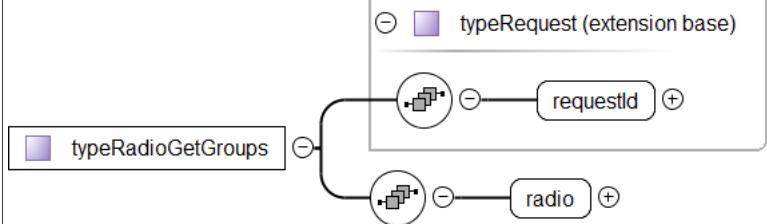
Namespace	DR-GW
Diagram	<pre> classDiagram typeRequest < -- typeRadioGet typeRequest "1..1" requestId typeRequest "1..1" radio typeRadioGet "1..1" requestId typeRadioGet "1..1" radio </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeRadioGet
Used by	Element interfaceRadio/get
Model	requestId , radio
Children	radio, requestId
Source	<pre> <xs:complexType name="typeRadioGet"> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress" /> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeRadioGetList

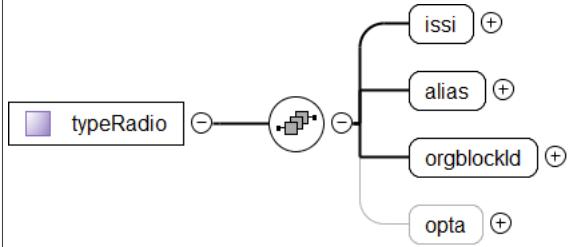
Namespace	DR-GW
Diagram	<pre> classDiagram typeRequest < -- typeRadioGetList typeRequest "1..1" requestId typeRequest "1..1" orgblockId typeRadioGetList "1..1" requestId typeRadioGetList "1..1" orgblockId </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeRadioGetList
Used by	Element interfaceRadio/getList
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId
Source	<pre> <xs:complexType name="typeRadioGetList"> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0" /> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeRadioGetGroups

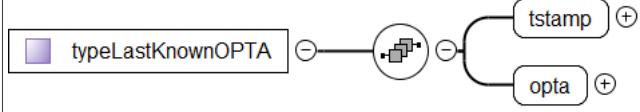
Namespace	DR-GW
-----------	-------

Diagram	
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeRadioGetGroups
Used by	Element interfaceRadio/getGroups
Model	requestId , radio
Children	radio, requestId
Source	<pre><xs:complexType name="typeRadioGetGroups"> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="radio" type="typeRadio"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeRadio

Namespace	DR-GW
Annotations	
Diagram	
Used by	Elements typeRadioEvent/radio, typeRadioGetEvent/radio, typeRadioGetGroups/radio, typeRadioGetListEvent/radio
Model	issi , alias , orgblockId , opta{0,1}
Children	alias, issi, opta, orgblockId
Source	<pre><xs:complexType name="typeRadio"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="issi" type="typeSubscriberAddress"/> <xs:element name="alias" type="xs:normalizedString"/> <xs:element name="orgblockId" type="typeOrganisationBlockId"/> <xs:element name="opta" type="typeLastKnownOPTA" minOccurs="0"/> </xs:sequence> </xs:complexType></pre>

Complex Type typeLastKnownOPTA

Namespace	DR-GW
Diagram	
Used by	Element typeRadio/opta

Model	tstamp , opta
Children	opta, tstamp
Source	<pre><xss:complexType name="typeLastKnownOPTA"> <xss:sequence> <xss:element name="tstamp" type="xs:dateTime"/> <xss:element name="opta" type="typeOpta"/> </xss:sequence> </xss:complexType></pre>

Complex Type typeRadioTrack

Namespace	DR-GW
Diagram	<pre> classDiagram typeRequest < -- typeRadioTrack typeRequest "0..1" -- "1..1" radio typeRequest "0..1" -- "1..1" stop typeRadioTrack "0..1" -- "1..1" radio typeRadioTrack "0..1" -- "1..1" stop </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeRadioTrack
Used by	Element interfaceRadio/track
Model	requestId , radio , stop
Children	radio, requestId, stop
Source	<pre><xss:complexType name="typeRadioTrack"> <xss:complexContent> <xss:extension base="typeRequest"> <xss:sequence> <xss:element name="radio" type="typeSubscriberAddress"/> <xss:element name="stop" type="xs:boolean"/> </xss:sequence> </xss:extension> </xss:complexContent> </xss:complexType></pre>

Complex Type typeRadioChangeOpta

Namespace	DR-GW
Diagram	<pre> classDiagram typeRequest < -- typeRadioChangeOpta typeRequest "0..1" -- "1..1" radio typeRequest "0..1" -- "1..1" opta typeRadioChangeOpta "0..1" -- "1..1" radio typeRadioChangeOpta "0..1" -- "1..1" opta </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeRadioChangeOpta
Used by	Element interfaceRadio/changeOpta
Model	requestId , radio , opta
Children	opta, radio, requestId
Source	<pre><xss:complexType name="typeRadioChangeOpta"></pre>

```

<xs:complexContent>
  <xs:extension base="typeRequest">
    <xs:sequence>
      <xs:element name="radio" type="typeSubscriberAddress"/>
      <xs:element name="opta" type="typeOpta"/>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>

```

Complex Type typeRadioEnable

Namespace	DR-GW
Annotations	This method is used to Enable the radio terminal over the air.
Diagram	<pre> classDiagram typeRequest < -- typeRadioEnable typeRequest { -requestId -radio } typeRadioEnable { +radio +reason +enable } </pre> <p>The diagram illustrates the inheritance relationship between the <code>typeRadioEnable</code> complex type and its base type, <code>typeRequest</code>. The <code>typeRequest</code> type is shown with a minus sign before its name, indicating it is an extension base. It contains two attributes: <code>requestId</code> (marked with a plus sign) and <code>radio</code> (marked with a plus sign). A line connects <code>typeRequest</code> to <code>typeRadioEnable</code>, which is also marked with a minus sign. <code>typeRadioEnable</code> contains three additional attributes: <code>radio</code> (marked with a plus sign), <code>reason</code> (marked with a plus sign), and <code>enable</code> (marked with a plus sign). A callout box provides the annotation: "This method is used to Enable the radio terminal over the air."</p>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeRadioEnable
Used by	Element interfaceRadio/enable
Model	requestId , radio , radio , reason{0,1} , enable
Children	enable, radio, reason, requestId
Source	<pre> <xs:complexType name="typeRadioEnable"> <xs:annotation> <xs:documentation>This method is used to Enable the radio terminal over the air.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="reason" type="xs:unsignedByte" minOccurs="0"/> <xs:element name="enable" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeRadioDisable

Namespace	DR-GW
Annotations	<p>This method is used to disable the radio terminal over the air. If no reason is supplied, then the DF-Gateway sets the default reason. There is no default reason value, it depends on the DF-Gateway configuration what reason is used when no reason is supplied by DF-Client.</p> <p>See TCS API Description for all possible reasons for disabling.</p>

Diagram	<pre> classDiagram typeRequest < -- typeRadioDisable typeRequest "0..1" requestId typeRadioDisable "0..1" radio typeRadioDisable "0..1" reason typeRadioDisable "0..1" enable </pre> <p>This method is used to disable the radio terminal over the air. If no reason is supplied, then the DF-Gateway sets the...</p>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeRadioDisable
Used by	Element interfaceRadio/disable
Model	requestId , radio , radio , reason{0,1} , enable
Children	enable, radio, reason, requestId
Source	<pre> <xs:complexType name="typeRadioDisable"> <xs:annotation> <xs:documentation>This method is used to disable the radio terminal over the air. If no reason is supplied, then the DF-Gateway sets the default reason. There is no default reason value, it depends on the DF-Gateway configuration what reason is used when no reason is supplied by DF-Client. See TCS API Description for all possible reasons for disabling.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress" /> <xs:element name="radio" type="typeSubscriberAddress" /> <xs:element name="reason" type="xs:unsignedByte" minOccurs="0" /> <xs:element name="enable" type="xs:boolean" /> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeRadioGetEvent

Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeRadioGetEvent typeEvent "0..1" requestId typeEvent "0..1" result typeRadioGetEvent "0..1" radio </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioGetEvent
Used by	Element interfaceRadio/getEvent
Model	requestId{0,1} , result{0,1} , radio
Children	radio, requestId, result
Source	<pre> <xs:complexType name="typeRadioGetEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress" /> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

```

<xs:element name="radio" type="typeRadio"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

Complex Type typeRadioGetListEvent

Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeRadioGetListEvent typeEvent { requestId result } typeRadioGetListEvent { radio * "0..infinity" listEnd } </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioGetListEvent
Used by	Element interfaceRadio/getListEvent
Model	requestId{0,1} , result{0,1} , radio* , listEnd
Children	listEnd, radio, requestId, result
Source	<pre> <xs:complexType name="typeRadioGetListEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeRadio" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="listEnd" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeRadioGetGroupsEvent

Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeRadioGetGroupsEvent typeEvent { requestId result } typeRadioGetGroupsEvent { radio group * "0..infinity" listEnd } </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioGetGroupsEvent
Used by	Element interfaceRadio/getGroupsEvent
Model	requestId{0,1} , result{0,1} , radio , group* , listEnd
Children	group, listEnd, radio, requestId, result

Source	<pre> <xs:complexType name="typeRadioGetGroupsEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="group" type="typeRadioGroupSelection" minOccurs="0" maxOccurs="unbounded" /> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>
--------	---

Complex Type typeRadioGroupSelection

Namespace	DR-GW
Annotations	
Diagram	<pre> graph LR typeEvent["typeEvent (extension base)"] typeRadioGroupSelection["typeRadioGroupSelection"] typeRadioGroupSelection --> group["group +"] typeRadioGroupSelection --> level["level +"] </pre>
Used by	Elements typeRadioGetGroupsEvent/group, typeRadioGroupsEvent/group
Model	group , level
Children	group, level
Source	<pre> <xs:complexType name="typeRadioGroupSelection"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> <xs:element name="level" type="typeGroupSelectionLevel"/> </xs:sequence> </xs:complexType> </pre>

Complex Type typeRadioTrackSubscriptionEvent

Namespace	DR-GW
Diagram	<pre> graph LR typeEvent["typeEvent (extension base)"] typeRadioTrackSubscriptionEvent["typeRadioTrackSubscriptionEvent"] typeRadioTrackSubscriptionEvent --> requestId["requestId +"] typeRadioTrackSubscriptionEvent --> result["result +"] typeRadioTrackSubscriptionEvent --> radio["radio +"] typeRadioTrackSubscriptionEvent --> stop["stop +"] </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioTrackSubscriptionEvent
Used by	Element interfaceRadio/trackSubscriptionEvent
Model	requestId{0,1} , result{0,1} , radio , stop
Children	radio, requestId, result, stop
Source	<pre> <xs:complexType name="typeRadioTrackSubscriptionEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="stop" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeRadioTrackEvent

Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeRadioTrackEvent typeEvent "0..1" *--> requestId typeEvent "0..1" *--> result typeRadioTrackEvent "0..1" *--> trackingData </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioTrackEvent
Used by	Element interfaceRadio/trackEvent
Model	requestId{0,1} , result{0,1} , trackingData
Children	requestId, result, trackingData
Source	<pre> <xss:complexType name="typeRadioTrackEvent"> <xss:complexContent> <xss:extension base="typeEvent"> <xss:sequence> <xss:element name="trackingData" type="typeRadioTrackingData"/> </xss:sequence> </xss:extension> </xss:complexContent> </xss:complexType> </pre>

Complex Type typeRadioTrackingData

Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeRadioTrackingData "0..1" *--> radio typeRadioTrackingData "0..1" *--> registered typeRadioTrackingData "0..1" *--> exchangeId typeRadioTrackingData "0..1" *--> locationArea typeRadioTrackingData "0..1" *--> lastActive typeRadioTrackingData "0..1" *--> scanningOn typeRadioTrackingData "0..1" *--> status typeRadioTrackingData "0..1" *--> callType typeRadioTrackingData "0..1" *--> callParty typeRadioTrackingData "0..1" *--> dmoState typeRadioTrackingData "0..1" *--> emergency </pre>
Used by	Element typeRadioTrackEvent/trackingData
Model	radio , registered , exchangeId , locationArea{0,1} , lastActive , scanningOn , status , callType , callParty , dmoState , emergency

Children	callParty, callType, dmoState, emergency, exchangeId, lastActive, locationArea, radio, registered, scanningOn, status
Source	<pre> <xs:complexType name="typeRadioTrackingData"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress" /> <xs:element name="registered" type="xs:boolean" /> <xs:element name="exchangeId" type="xs:unsignedLong" /> <xs:element name="locationArea" type="xs:unsignedShort" minOccurs="0" /> <xs:element name="lastActive" type="xs:dateTime" /> <xs:element name="scanningOn" type="xs:boolean" /> <xs:element name="status" type="typeStatusIndicator" /> <xs:element name="callType" type="typeCallType" /> <xs:element name="callParty" type="typeSubscriberAddress" /> <xs:element name="dmoState" type="xs:boolean" /> <xs:element name="emergency" type="xs:boolean" /> </xs:sequence> </xs:complexType></pre>

Complex Type typeStatusIndicator

Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeEvent < -- typeStatusIndicator typeStatusIndicator { <<xs:element name="value" type="xs:unsignedLong"/>> <<xs:element name="time" type="xs:dateTime"/>> } </pre>
Used by	Element typeRadioTrackingData/status
Model	value , time
Children	time, value
Source	<pre> <xs:complexType name="typeStatusIndicator"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="value" type="xs:unsignedLong" /> <xs:element name="time" type="xs:dateTime" /> </xs:sequence> </xs:complexType></pre>

Complex Type typeRadioGroupsEvent

Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeRadioGroupsEvent typeRadioGroupsEvent { <<xs:element name="requestId" type="xs:string"/>> <<xs:element name="result" type="xs:string"/>> <<xs:element name="radio" type="typeRadio"/>> <<xs:element name="group" type="group" minOccurs="1..infinity"/>> <<xs:element name="deletedGroup" type="group" />> } </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioGroupsEvent
Used by	Element interfaceRadio/groupsEvent
Model	requestId{0,1} , result{0,1} , radio , (group+ deletedGroup)
Children	deletedGroup, group, radio, requestId, result

Source	<pre><xs:complexType name="typeRadioGroupsEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:choice> <xs:element name="group" type="typeRadioGroupSelection" maxOccurs="unbounded"/> <xs:element name="deletedGroup" type="typeSubscriberAddress"/> </xs:choice> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>
--------	---

Complex Type typeRadioChangeOptaEvent

Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeRadioChangeOptaEvent typeEvent { +requestId +result } typeRadioChangeOptaEvent { +radio +opta } </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent <ul style="list-style-type: none"> • typeRadioChangeOptaEvent
Used by	Element interfaceRadio/changeOptaEvent
Model	requestId{0,1} , result{0,1} , radio , opta
Children	opta, radio, requestId, result
Source	<pre><xs:complexType name="typeRadioChangeOptaEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="opta" type="typeOpta"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeRadioEnableDisableEvent

Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeRadioEnableDisableEvent typeEvent { +requestId +result } typeRadioEnableDisableEvent { +radio +reason +enabled +overTheAir } </pre>

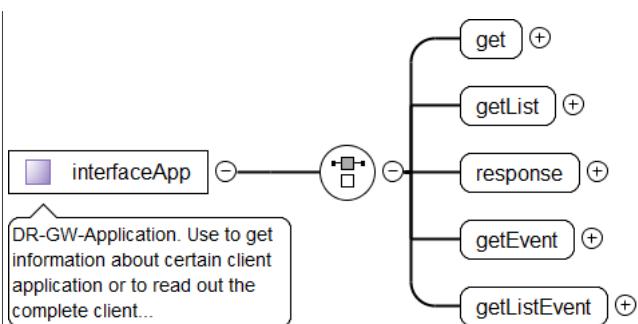
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioEnableDisableEvent
Used by	Element interfaceRadio/enableDisableEvent
Model	requestId{0,1} , result{0,1} , radio , reason{0,1} , enabled , overTheAir{0,1}
Children	enabled, overTheAir, radio, reason, requestId, result
Source	<pre><xs:complexType name="typeRadioEnableDisableEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress" /> <xs:element name="reason" type="xs:unsignedByte" minOccurs="0" /> <xs:element name="enabled" type="xs:boolean" /> <xs:element name="overTheAir" type="xs:boolean" minOccurs="0" /> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeRadioEvent

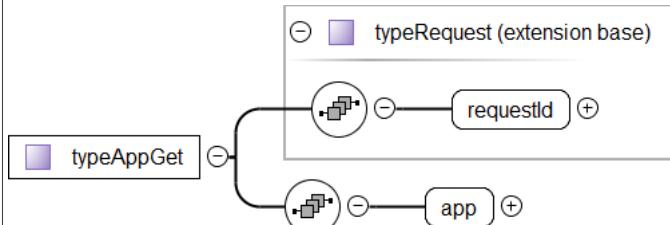
Namespace	DR-GW
Diagram	<pre> classDiagram typeEvent < -- typeRadioEvent typeEvent { -requestId -result } typeRadioEvent { -radio -delete } </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeRadioEvent
Used by	Element interfaceRadio/event
Model	requestId{0,1} , result{0,1} , radio , delete
Children	delete, radio, requestId, result
Source	<pre><xs:complexType name="typeRadioEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeRadio" /> <xs:element name="delete" type="xs:boolean" /> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type interfaceApp

Namespace	DR-GW
Annotations	DR-GW-Application. Use to get information about certain client application or to read out the complete client application list the DF-Client has rights for. Use only via SOAP.

Diagram	
Used by	Element drgw/app
Model	get getList response getEvent getListEvent
Children	get, getEvent, getList, getListEvent, response
Source	<pre><xs:complexType name="interfaceApp"> <xs:annotation> <xs:documentation>DR-GW-Application. Use to get information about certain client application or to read out the complete client application list the DF-Client has rights for. Use only via SOAP.</xs:documentation> </xs:annotation> <xs:choice> <xs:element name="get" type="typeAppGet"/> <xs:element name="getList" type="typeAppGetList"/> <xs:element name="response" type="typeResponse"/> <xs:element name="getEvent" type="typeAppGetEvent"/> <xs:element name="getListEvent" type="typeAppGetListEvent"/> </xs:choice> </xs:complexType></pre>

Complex Type typeAppGet

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeAppGet
Used by	Element interfaceApp/get
Model	requestId , app
Children	app, requestId
Source	<pre><xs:complexType name="typeAppGet"> <xs:annotation> <xs:documentation></xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="app" type="typeSubscriberAddress"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeAppGetList

Namespace	DR-GW
-----------	-------

Annotations	
Diagram	<pre> classDiagram typeRequest < -- typeAppGetList typeRequest "0..1" requestId typeRequest "0..1" orgblockId typeAppGetList "0..1" app </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeAppGetList
Used by	Element interfaceApp/getList
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId
Source	<pre> <xss:complexType name="typeAppGetList"> <xss:annotation> <xss:documentation></xss:documentation> </xss:annotation> <xss:complexContent> <xss:extension base="typeRequest"> <xss:sequence> <xss:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0"/> </xss:sequence> </xss:extension> </xss:complexContent> </xss:complexType> </pre>

Complex Type typeAppGetEvent

Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeEvent < -- typeAppGetEvent typeEvent "0..1" requestId typeEvent "0..1" result typeAppGetEvent "0..1" app </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeAppGetEvent
Used by	Element interfaceApp/getEvent
Model	requestId{0,1} , result{0,1} , app
Children	app, requestId, result
Source	<pre> <xss:complexType name="typeAppGetEvent"> <xss:annotation> <xss:documentation></xss:documentation> </xss:annotation> <xss:complexContent> <xss:extension base="typeEvent"> <xss:sequence> <xss:element name="app" type="typeApplication" maxOccurs="1"/> </xss:sequence> </xss:extension> </xss:complexContent> </xss:complexType> </pre>

Complex Type typeApplication

Namespace	DR-GW
Annotations	
Diagram	<pre> graph LR typeEvent["typeEvent (extension base)"] typeApplication["typeApplication"] typeEvent --> typeApplication typeApplication --- addr["addr +"] typeApplication --- alias["alias +"] typeApplication --- orgblockId["orgblockId +"] </pre>
Used by	Elements typeAppGetEvent/app, typeAppGetListEvent/app
Model	addr, alias, orgblockId
Children	addr, alias, orgblockId
Source	<pre> <xs:complexType name="typeApplication"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="addr" type="typeSubscriberAddress"/> <xs:element name="alias" type="xs:normalizedString"/> <xs:element name="orgblockId" type="typeOrganisationBlockId"/> </xs:sequence> </xs:complexType> </pre>

Complex Type typeAppGetListEvent

Namespace	DR-GW
Annotations	
Diagram	<pre> graph LR typeEvent["typeEvent (extension base)"] typeAppGetListEvent["typeAppGetListEvent"] typeEvent --> typeAppGetListEvent typeAppGetListEvent --- requestId["requestId +"] typeAppGetListEvent --- result["result +"] typeAppGetListEvent --- app["app +"] typeAppGetListEvent --- listEnd["listEnd +"] </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeAppGetListEvent
Used by	Element interfaceApp/getListEvent
Model	requestId{0,1}, result{0,1}, app*, listEnd{0,1}
Children	app, listEnd, requestId, result
Source	<pre> <xs:complexType name="typeAppGetListEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="app" type="typeApplication" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="listEnd" type="xs:boolean" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type interfaceSystem

Namespace	DR-GW
-----------	-------

Annotations	DR-GW-System. Use to get TETRA system events or DR-GW vendor-specific log and/or system events. Use only via SOAP.
Diagram	<pre> classDiagram class interfaceSystem class tetraStatesEvent class logEvent class event interfaceSystem < -- tetraStatesEvent interfaceSystem < -- logEvent interfaceSystem < -- event </pre> <p>DR-GW-System. Use to get TETRA system events or DR-GW vendor-specific log and/or system events. Use only via SOAP.</p>
Used by	Element drgw/system
Model	tetraStatesEvent logEvent event
Children	event, logEvent, tetraStatesEvent
Source	<pre> <xs:complexType name="interfaceSystem"> <xs:annotation> <xs:documentation>DR-GW-System. Use to get TETRA system events or DR-GW vendor-specific log and/or system events. Use only via SOAP.</xs:documentation> </xs:annotation> <xs:choice> <xs:element name="tetraStatesEvent" type="typeSystemTetraStatesEvent"/> <xs:element name="logEvent" type="typeSystemLogEvent"/> <xs:element name="event" type="typeSystemEvent"/> </xs:choice> </xs:complexType> </pre>

Complex Type typeSystemTetraStatesEvent

Namespace	DR-GW
Annotations	Indication of the subsystem state updates. Contains current states of all subsystem known by the TCS.
Diagram	<pre> classDiagram class typeEvent { <<extension base>> } class typeSystemTetraStatesEvent { <<extension base>> } typeEvent < -- typeSystemTetraStatesEvent typeSystemTetraStatesEvent < -- requestId typeSystemTetraStatesEvent < -- result typeSystemTetraStatesEvent < -- tcsState typeSystemTetraStatesEvent < -- dxtState typeSystemTetraStatesEvent < -- cddconnectionState typeSystemTetraStatesEvent < -- cddserverState </pre> <p>Indication of the subsystem state updates. Contains current states of all subsystem known by the TCS.</p>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeSystemTetraStatesEvent
Used by	Element interfaceSystem/tetraStatesEvent
Model	requestId{0,1} , result{0,1} , tcsState{0,1} , dxtState{0,1} , cddconnectionState{0,1} , cddserverState{0,1}
Children	cddconnectionState, cddserverState, dxtState, requestId, result, tcsState
Source	<pre> <xs:complexType name="typeSystemTetraStatesEvent"> <xs:annotation> <xs:documentation>Indication of the subsystem state updates. Contains current states of all subsystem known by the TCS.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="tcsState" type="typeSystemElementState" minOccurs="0"/> <xs:element name="dxtState" type="typeSystemElementState" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

```

<xs:element name="cddconnectionState" type="typeSystemElementState" minOccurs="0"/>
<xs:element name="cddserverState" type="typeSystemElementState" minOccurs="0"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

Complex Type typeSystemLogEvent

Namespace	DR-GW
Annotations	DR-GW vendor-specific logging information (errors, notices) in both numeric and textual form. See DR-GW vendor-specific documentation for the details.
Diagram	<pre> classDiagram typeEvent < -- typeSystemLogEvent typeEvent : typeEvent (extension base) typeEvent --> requestId : requestId + (optional) typeEvent --> result : result + (optional) typeEvent --> value : value + (optional) typeEvent --> text : text + (optional) typeSystemLogEvent : typeSystemLogEvent typeSystemLogEvent --> DR-GW vendor-specific logging information (errors, notices) in both numeric and textual form. See DR-GW vendor-specific... </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeSystemLogEvent
Used by	Element interfaceSystem/logEvent
Model	requestId{0,1} , result{0,1} , value{0,1} , text{0,1}
Children	requestId, result, text, value
Source	<pre> <xs:complexType name="typeSystemLogEvent"> <xs:annotation> <xs:documentation>DR-GW vendor-specific logging information (errors, notices) in both numeric and textual form. See DR-GW vendor-specific documentation for the details.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="value" type="xs:hexBinary" minOccurs="0"/> <xs:element name="text" type="xs:normalizedString" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeSystemEvent

Namespace	DR-GW
Annotations	DR-GW vendor-specific system information in both numeric and textual form. See DR-GW vendor-specific documentation for the details.
Diagram	<pre> classDiagram typeEvent < -- typeSystemEvent typeEvent : typeEvent (extension base) typeEvent --> requestId : requestId + (optional) typeEvent --> result : result + (optional) typeEvent --> value : value + (optional) typeEvent --> text : text + (optional) typeSystemEvent : typeSystemEvent typeSystemEvent --> DR-GW vendor-specific system information in both numeric and textual form. See DR-GW vendor-specific... </pre>

Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeSystemEvent
Used by	Element interfaceSystem/event
Model	requestId{0,1} , result{0,1} , value{0,1} , text{0,1}
Children	requestId, result, text, value
Source	<pre><xs:complexType name="typeSystemEvent"> <xs:annotation> <xs:documentation>DR-GW vendor-specific system information in both numeric and textual form. See DR-GW vendor-specific documentation for the details.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="value" type="xs:hexBinary" minOccurs="0"/> <xs:element name="text" type="xs:normalizedString" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeCallRequest

Namespace	DR-GW
Annotations	The method is used to accomplish all call related operations.
Diagram	<pre> classDiagram typeRequest("typeRequest (extension base)") { requestId action attributes callingParty calledParty workstationId } typeCallRequest("typeCallRequest") { <<The method is used to accomplish all call related operations.>> } typeRequest < -- typeCallRequest typeCallRequest --o typeRequest : requestId typeCallRequest --o typeRequest : action typeCallRequest --o typeRequest : attributes typeCallRequest --o typeRequest : callingParty typeCallRequest --o typeRequest : calledParty typeCallRequest --o typeRequest : workstationId </pre>
Type	extension of typeRequest
Type hierarchy	<ul style="list-style-type: none"> • typeRequest • typeCallRequest
Model	requestId , action , attributes{0,1} , callingParty{0,1} , calledParty{0,1} , workstationId{0,1}
Children	action, attributes, calledParty, callingParty, requestId, workstationId
Source	<pre><xs:complexType name="typeCallRequest"> <xs:annotation> <xs:documentation>The method is used to accomplish all call related operations.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="action" type="typeActionRequest"/> <xs:element name="attributes" type="typeCallAttributes" minOccurs="0"/> <xs:element name="callingParty" type="typeAddress" minOccurs="0"/> <xs:element name="calledParty" type="typeAddress" minOccurs="0"/> <xs:element name="workstationId" type="typeWorkstationId" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType></pre>

Complex Type typeSessionLogoutEvent

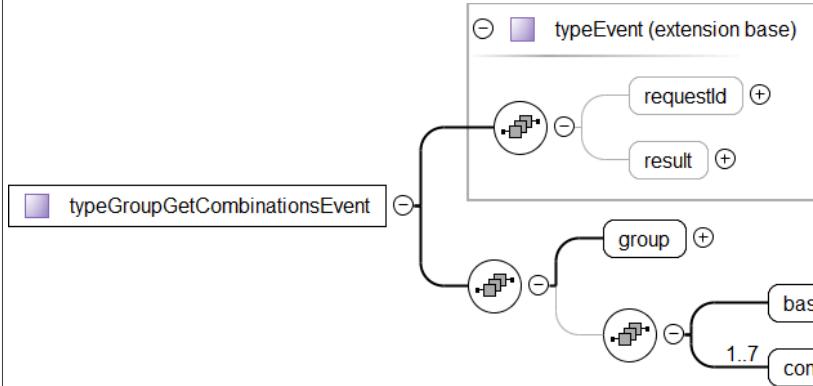
Namespace	DR-GW
Annotations	
Diagram	<pre> classDiagram typeEvent < -- typeSessionLogoutEvent typeEvent "0..1" *--> requestId typeEvent "0..1" *--> result typeEvent "0..1" *--> reason </pre>
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent <ul style="list-style-type: none"> • typeSessionLogoutEvent
Model	requestId{0,1} , result{0,1} , reason{0,1}
Children	reason, requestId, result
Source	<pre> <xs:complexType name="typeSessionLogoutEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="reason" type="xs:unsignedLong" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> </pre>

Complex Type typeSdsValidity

Namespace	DR-GW
Annotations	Validity of the SDS in case store and forward center is used. The unit is seconds. Infinte validity is represented by 0xFFFFFFFF
Diagram	<p>Validity of the SDS in case store and forward center is used. The unit is seconds. Infinte validity is represented by...</p>
Model	value
Children	value
Source	<pre> <xs:complexType name="typeSdsValidity"> <xs:annotation> <xs:documentation>Validity of the SDS in case store and forward center is used. The unit is seconds. Infinte validity is represented by 0xFFFFFFFF</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="value" type="xs:unsignedLong"/> </xs:sequence> </xs:complexType> </pre>

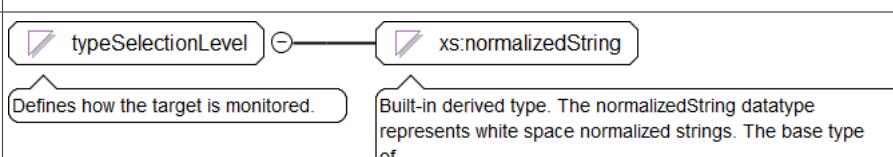
Complex Type typeGroupGetCombinationsEvent

Namespace	DR-GW
Annotations	

Diagram	
Type	extension of typeEvent
Type hierarchy	<ul style="list-style-type: none"> • typeEvent • typeGroupGetCombinationsEvent
Model	requestId{0,1} , result{0,1} , group , baseGroup , constitGroup{1,7}
Children	baseGroup, constitGroup, group, requestId, result
Source	<pre><xss:complexType name="typeGroupGetCombinationsEvent"> <xss:annotation> <xss:documentation/> </xss:annotation> <xss:complexContent> <xss:extension base="typeEvent"> <xss:sequence> <xss:element name="group" type="typeSubscriberAddress" /> <xss:sequence minOccurs="0"> <xss:element name="baseGroup" type="typeSubscriberAddress" /> <xss:element name="constitGroup" type="typeSubscriberAddress" maxOccurs="7" /> </xss:sequence> </xss:sequence> </xss:extension> </xss:complexContent> </xss:complexType></pre>

Simple Type(s)

Simple Type typeSelectionLevel

Namespace	DR-GW															
Annotations	Defines how the target is monitored.															
Diagram																
Type	restriction of xs:normalizedString															
Facets	<table border="1"> <tr> <td>enumeration</td> <td>no</td> <td>No selection. Used to remove selection.</td> </tr> <tr> <td>enumeration</td> <td>event</td> <td>Event monitoring.</td> </tr> <tr> <td>enumeration</td> <td>audio</td> <td>Audio monitoring.</td> </tr> <tr> <td>enumeration</td> <td>use</td> <td>Selection level use.</td> </tr> <tr> <td>enumeration</td> <td>a_use</td> <td>Selection level active use.</td> </tr> </table>	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.
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	<pre><xss:simpleType name="typeSelectionLevel"> <xss:annotation> <xss:documentation>Defines how the target is monitored.</xss:documentation> </xss:annotation> <xss:restriction base="xs:normalizedString"> <xss:enumeration value="no"> <xss:annotation> <xss:documentation>No selection. Used to remove selection.</xss:documentation> </xss:annotation> </xss:enumeration> </xss:restriction> </xss:simpleType></pre>															

```

</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 typeDialString

Namespace	DR-GW
Annotations	Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.
Diagram	<p>The diagram shows two boxes: 'typeDialString' on the left and 'xs:normalizedString' on the right. An association line connects them with a hollow circle at the 'typeDialString' end. Two callout boxes point to these elements: one pointing to 'typeDialString' stating 'Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.' and another pointing to 'xs:normalizedString' stating 'Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...'.</p>
Type	restriction of xs:normalizedString
Facets	maxLength 24
Used by	Elements typeAddress/msisdn, typeExternal/number
Source	<pre> <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> </pre>

Simple Type typeOpta

Namespace	DR-GW
Annotations	OPTA string. Maximum length is 24 characters.
Diagram	<p>The diagram shows two boxes: 'typeOpta' on the left and 'xs:normalizedString' on the right. An association line connects them with a hollow circle at the 'typeOpta' end. Two callout boxes point to these elements: one pointing to 'typeOpta' stating 'OPTA string. Maximum length is 24 characters.' and another pointing to 'xs:normalizedString' stating 'Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...'.</p>
Type	restriction of xs:normalizedString
Facets	maxLength 24
Used by	Elements typeAddress/opta, typeLastKnownOPTA/opta, typeRadioChangeOpta/opta, typeRadioChangeOptaEvent/opta
Source	<pre> <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> </pre>

```

    </xs:restriction>
</xs:simpleType>

```

Simple Type typeResponseCode

Namespace	DR-GW												
Diagram	<p>A UML class diagram showing a derived type relationship. A box labeled "typeResponseCode" has a line with a hollow circle at its end pointing to a box labeled "xs:normalizedString". A callout box points to the "xs:normalizedString" box with the text: "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...".</p>												
Type	restriction of xs:normalizedString												
Facets	<table> <tr> <td>enumeration</td> <td>success</td> </tr> <tr> <td>enumeration</td> <td>final_response_pending</td> </tr> <tr> <td>enumeration</td> <td>error</td> </tr> <tr> <td>enumeration</td> <td>not_authorized_error</td> </tr> <tr> <td>enumeration</td> <td>temporary_failure</td> </tr> <tr> <td>enumeration</td> <td>subscription_failed</td> </tr> </table>	enumeration	success	enumeration	final_response_pending	enumeration	error	enumeration	not_authorized_error	enumeration	temporary_failure	enumeration	subscription_failed
enumeration	success												
enumeration	final_response_pending												
enumeration	error												
enumeration	not_authorized_error												
enumeration	temporary_failure												
enumeration	subscription_failed												
Used by	Element typeResult/responseCode												
Source	<pre> <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> </pre>												

Simple Type typeSourceSystem

Namespace	DR-GW						
Diagram	<p>A UML class diagram showing a derived type relationship. A box labeled "typeSourceSystem" has a line with a hollow circle at its end pointing to a box labeled "xs:normalizedString". A callout box points to the "xs:normalizedString" box with the text: "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...".</p>						
Type	restriction of xs:normalizedString						
Facets	<table> <tr> <td>enumeration</td> <td>DR-GW</td> </tr> <tr> <td>enumeration</td> <td>TCS-API</td> </tr> <tr> <td>enumeration</td> <td>TETRA</td> </tr> </table>	enumeration	DR-GW	enumeration	TCS-API	enumeration	TETRA
enumeration	DR-GW						
enumeration	TCS-API						
enumeration	TETRA						
Used by	Element typeResult/sourceSystem						
Source	<pre> <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> </pre>						

Simple Type type.ActionEvent

Namespace	DR-GW
Annotations	All possible call actions.
Diagram	<p>A UML class diagram showing a derived type relationship. A box labeled "type.ActionEvent" has a line with a hollow circle at its end pointing to a box labeled "xs:normalizedString". Two callout boxes point to the "xs:normalizedString" box with the text: "All possible call actions." and "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...".</p>

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	typeCallEvent/action	
Source	<pre><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></pre>		

Simple Type typeCallMode

Namespace	DR-GW
Annotations	Call mode attribute. Choices are simplex or duplex.
Diagram	<pre> classDiagram typeCallMode < -- xs:normalizedString </pre> <p>The diagram shows a UML class hierarchy. A box labeled "typeCallMode" has a generalization arrow pointing to another box labeled "xs:normalizedString". Below the boxes, two callouts provide additional information: one states "Call mode attribute. Choices are simplex or duplex.", and the other states "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...".</p>
Type	restriction of xs:normalizedString
Facets	enumeration simplex

	enumeration	duplex
Used by	Element	typeCallAttributes/mode
Source	<pre><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></pre>	

Simple Type typeCallType

Namespace	DR-GW						
Annotations	Call type attribute. Choices are Point2Point, Point2MultiPoint or Broadcast.						
Diagram	<p>The diagram shows a UML class named "typeCallType" with a hollow circle symbol to its right, indicating it is a derived type. A line connects "typeCallType" to another class named "xs:normalizedString". Two callouts point to these elements: one pointing to "typeCallType" with the text "Call type attribute. Choices are Point2Point, Point2MultiPoint or Broadcast.", and another pointing to "xs:normalizedString" with the text "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...".</p>						
Type	restriction of xs:normalizedString						
Facets	<table border="1"> <tr> <td>enumeration</td> <td>p2p</td> </tr> <tr> <td>enumeration</td> <td>p2mp</td> </tr> <tr> <td>enumeration</td> <td>bcast</td> </tr> </table>	enumeration	p2p	enumeration	p2mp	enumeration	bcast
enumeration	p2p						
enumeration	p2mp						
enumeration	bcast						
Used by	Elements typeCallAttributes/commtype, typeRadioTrackingData/callType						
Source	<pre><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></pre>						

Simple Type typeTxDemandPriority

Namespace	DR-GW						
Annotations	Defines priority of speech item request: normal, pre-emptive, or emergency.						
Diagram	<p>The diagram shows a UML class named "typeTxDemandPriority" with a hollow circle symbol to its right, indicating it is a derived type. A line connects "typeTxDemandPriority" to another class named "xs:normalizedString". Two callouts point to these elements: one pointing to "typeTxDemandPriority" with the text "Defines priority of speech item request: normal, pre-emptive, or emergency.", and another pointing to "xs:normalizedString" with the text "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...".</p>						
Type	restriction of xs:normalizedString						
Facets	<table border="1"> <tr> <td>enumeration</td> <td>normal</td> </tr> <tr> <td>enumeration</td> <td>preemptive</td> </tr> <tr> <td>enumeration</td> <td>emergency</td> </tr> </table>	enumeration	normal	enumeration	preemptive	enumeration	emergency
enumeration	normal						
enumeration	preemptive						
enumeration	emergency						
Used by	Element typeCallAttributes/demandPriority						
Source	<pre><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:restriction> </xs:simpleType></pre>						

```

<xs:enumeration value="preemptive"/>
<xs:enumeration value="emergency"/>
</xs:restriction>
</xs:simpleType>

```

Simple Type typeActionPTTRequest

Namespace	DR-GW								
Annotations	All possible PTT requests.								
Diagram	<pre> classDiagram typeActionPTTRequest < -- xs:normalizedString </pre> <p>All possible PTT requests.</p> <p>Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...</p>								
Type	restriction of xs:normalizedString								
Facets	<table> <tr> <td>enumeration</td> <td>demandtx</td> <td>This method can be used to request a speech item for a connected call.</td> </tr> <tr> <td>enumeration</td> <td>ceasetx</td> <td>This method is used to inform the system that the speech item is not needed any more.</td> </tr> </table>			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.
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.							
Used by	Element typeCallPTTRequest/action								
Source	<pre> <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> </pre>								

Simple Type typeWorkstationId

Namespace	DR-GW		
Annotations	Optional parameter is used to support the "neighbours" feature.		
Diagram	<pre> classDiagram typeWorkstationId < -- xs:normalizedString </pre> <p>Optional parameter is used to support the "neighbours" feature.</p> <p>Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...</p>		
Type	xs:normalizedString		
Used by	Elements typeCallPTTRequest/workstationId, typeCallRequest/workstationId		
Source	<pre> <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:simpleType> </pre>		

Simple Type typeKeyExchangeAction

Namespace	DR-GW		
Annotations	Action type for key exchange request.		

Diagram	<pre> graph LR A["typeKeyExchangeAction"] -- ⊂ --> B["xs:normalizedString"] A --- C["Action type for key exchange request."] B --- D["Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of..."] </pre>				
Type	restriction of xs:normalizedString				
Facets	<table> <tr> <td>enumeration</td><td>start</td></tr> <tr> <td>enumeration</td><td>stop</td></tr> </table>	enumeration	start	enumeration	stop
enumeration	start				
enumeration	stop				
Used by	Element typeCallKeyExchange/action				
Source	<pre> <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> </pre>				

Simple Type typeTxGrant

Namespace	DR-GW								
Annotations	Defines to whom speech item was granted.								
Diagram	<pre> graph LR A["typeTxGrant"] -- ⊂ --> B["xs:normalizedString"] A --- C["Defines to whom speech item was granted."] B --- D["Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of..."] </pre>								
Type	restriction of xs:normalizedString								
Facets	<table> <tr> <td>enumeration</td><td>granted</td></tr> <tr> <td>enumeration</td><td>notGranted</td></tr> <tr> <td>enumeration</td><td>queued</td></tr> <tr> <td>enumeration</td><td>granted2another</td></tr> </table>	enumeration	granted	enumeration	notGranted	enumeration	queued	enumeration	granted2another
enumeration	granted								
enumeration	notGranted								
enumeration	queued								
enumeration	granted2another								
Used by	Element typeTxGranted/txGrant								
Source	<pre> <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> </pre>								

Simple Type typeTxPriority

Namespace	DR-GW				
Annotations	Defines the priority of the transmission.				
Diagram	<pre> graph LR A["typeTxPriority"] -- ⊂ --> B["xs:normalizedString"] A --- C["Defines the priority of the transmission."] B --- D["Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of..."] </pre>				
Type	restriction of xs:normalizedString				
Facets	<table> <tr> <td>enumeration</td><td>normal</td></tr> <tr> <td>enumeration</td><td>emergency</td></tr> </table>	enumeration	normal	enumeration	emergency
enumeration	normal				
enumeration	emergency				

Used by	Element	typeTxGranted/txPriority
Source		<pre><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></pre>

Simple Type typeUnitInEmergencyType

Namespace	DR-GW										
Annotations	Defines type of the subscriber. Refer to type tcsCallSubscriberType_t of the TCS-API.										
Diagram	<pre> classDiagram typeUnitInEmergencyType < -- xs:normalizedString note over typeUnitInEmergencyType: Defines type of the subscriber. Refer to type tcsCallSubscriberType_t of the TCS-API. note over xs:normalizedString: Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of... </pre>										
Type	restriction of xs:normalizedString										
Facets	<table border="1"> <tr><td>enumeration</td><td>dummy</td></tr> <tr><td>enumeration</td><td>ms</td></tr> <tr><td>enumeration</td><td>g4wif</td></tr> <tr><td>enumeration</td><td>external</td></tr> <tr><td>enumeration</td><td>ws</td></tr> </table>	enumeration	dummy	enumeration	ms	enumeration	g4wif	enumeration	external	enumeration	ws
enumeration	dummy										
enumeration	ms										
enumeration	g4wif										
enumeration	external										
enumeration	ws										
Used by	Element										
Source	<pre><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></pre>										

Simple Type typeEmergencyInfo

Namespace	DR-GW												
Annotations	Defines action taken by user in emergency.												
Diagram	<pre> classDiagram typeEmergencyInfo < -- xs:normalizedString note over typeEmergencyInfo: Defines action taken by user in emergency. note over xs:normalizedString: Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of... </pre>												
Type	restriction of xs:normalizedString												
Facets	<table border="1"> <tr><td>enumeration</td><td>addTx</td></tr> <tr><td>enumeration</td><td>add</td></tr> <tr><td>enumeration</td><td>ceased</td></tr> <tr><td>enumeration</td><td>demandTx</td></tr> <tr><td>enumeration</td><td>removed</td></tr> <tr><td>enumeration</td><td>emergencyCallDisconnected</td></tr> </table>	enumeration	addTx	enumeration	add	enumeration	ceased	enumeration	demandTx	enumeration	removed	enumeration	emergencyCallDisconnected
enumeration	addTx												
enumeration	add												
enumeration	ceased												
enumeration	demandTx												
enumeration	removed												
enumeration	emergencyCallDisconnected												
Used by	Element												
Source	<pre><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></pre>												

Source	<pre><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></pre>
--------	---

Simple Type typeKeyExchangeState

Namespace	DR-GW										
Annotations	Represents current key state.										
Diagram	<pre> classDiagram typeKeyExchangeState < -- xs:normalizedString note over typeKeyExchangeState: Represents current key state. note over xs:normalizedString: Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of... </pre>										
Type	restriction of xs:normalizedString										
Facets	<table border="1"> <tr> <td>enumeration</td> <td>keyValid</td> <td>current key is valid, no user action required.</td> </tr> <tr> <td>enumeration</td> <td>keyInvalid</td> <td>Key invalid, user must request key exchange.</td> </tr> <tr> <td>enumeration</td> <td>keyExchangeInProgress</td> <td>Key exchange in progress, user may abort exchange or wait until it gets finished.</td> </tr> </table>		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.
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 typeCallKeyExchangeEvent/state										
Source	<pre><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></pre>										

Simple Type typeKeyExchangeCode

Namespace	DR-GW	
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	<pre> classDiagram typeKeyExchangeCode < -- xs:hexBinary note over typeKeyExchangeCode: See "Table 5.3: Status words of the commands" of the E-to-E Encryption SIM-ME Interface (Version 4.0.5) for all... note over xs:hexBinary: Built-in primitive type. The hexBinary datatype represents arbitrary hex-encoded binary data. </pre>	
Type	restriction of xs:hexBinary	
Facets	length 2	

Used by	Element typeCallKeyExchangeEvent/code
Source	<pre><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></pre>

Simple Type typeKeyExchangeTextPriority

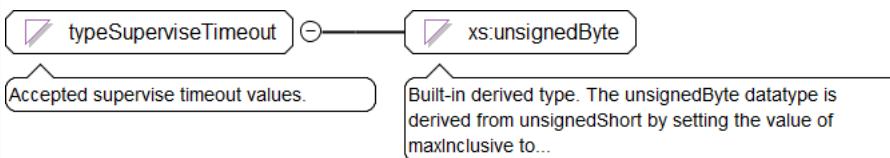
Namespace	DR-GW				
Annotations	Defines the priority of the KeyExchange information.				
Diagram	<pre> classDiagram typeKeyExchangeTextPriority < -- xs:normalizedString </pre> <p>The diagram shows a UML class named 'typeKeyExchangeTextPriority' with a generalization arrow pointing to the built-in type 'xs:normalizedString'. A callout box below the class definition states: 'Defines the priority of the KeyExchange information'. Another callout box below the inheritance arrow states: 'Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...'.</p>				
Type	restriction of xs:normalizedString				
Facets	<table border="1"> <tr> <td>enumeration</td> <td>normal</td> </tr> <tr> <td>enumeration</td> <td>high</td> </tr> </table>	enumeration	normal	enumeration	high
enumeration	normal				
enumeration	high				
Used by	Element typeCallKeyExchangeEvent/priority				
Source	<pre><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></pre>				

Simple Type typeKeyExchangeText

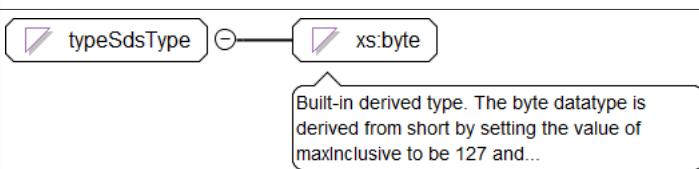
Namespace	DR-GW
Annotations	The textual information supplied by the BOS-simcard and sent from the DF-Gateway to the DF-client.
Diagram	<pre> classDiagram typeKeyExchangeText < -- xs:normalizedString </pre> <p>The diagram shows a UML class named 'typeKeyExchangeText' with a generalization arrow pointing to the built-in type 'xs:normalizedString'. A callout box below the class definition states: 'The textual information supplied by the BOS-simcard and sent from the DF-Gateway to the DF-client'. Another callout box below the inheritance arrow states: 'Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...'.</p>
Type	restriction of xs:normalizedString
Facets	maxLength 100
Used by	Element typeCallKeyExchangeEvent/text
Source	<pre><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></pre>

Simple Type typeSuperviseTimeout

Namespace	DR-GW
Annotations	Accepted supervise timeout values.

Diagram							
Type	restriction of xs:unsignedByte						
Facets	<table> <tr> <td>enumeration</td><td>20</td></tr> <tr> <td>enumeration</td><td>30</td></tr> <tr> <td>enumeration</td><td>60</td></tr> </table>	enumeration	20	enumeration	30	enumeration	60
enumeration	20						
enumeration	30						
enumeration	60						
Used by	Element typeSessionLogin/supervise						
Source	<pre><xs:simpleType name="typeSuperviseTimeout"> <xs:annotation> <xs:documentation>Accepted supervise timeout values.</xs:documentation> </xs:annotation> <xs:restriction base="xs:unsignedByte"> <xs:enumeration value="20"/> <xs:enumeration value="30"/> <xs:enumeration value="60"/> </xs:restriction> </xs:simpleType></pre>						

Simple Type typeSdsType

Namespace	DR-GW																		
Annotations																			
Diagram																			
Type	restriction of xs:byte																		
Facets	<table> <tr> <td>enumeration</td><td>0</td><td>SDS1.</td></tr> <tr> <td>enumeration</td><td>1</td><td>SDS2.</td></tr> <tr> <td>enumeration</td><td>2</td><td>SDS3.</td></tr> <tr> <td>enumeration</td><td>3</td><td>SDS4.</td></tr> <tr> <td>enumeration</td><td>4</td><td>SDS-TL.</td></tr> <tr> <td>enumeration</td><td>5</td><td>Status.</td></tr> </table>	enumeration	0	SDS1.	enumeration	1	SDS2.	enumeration	2	SDS3.	enumeration	3	SDS4.	enumeration	4	SDS-TL.	enumeration	5	Status.
enumeration	0	SDS1.																	
enumeration	1	SDS2.																	
enumeration	2	SDS3.																	
enumeration	3	SDS4.																	
enumeration	4	SDS-TL.																	
enumeration	5	Status.																	
Used by	Element typeSds/sdsType																		
Source	<pre><xs:simpleType name="typeSdsType"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:restriction base="xs:byte"> <xs:enumeration value="0"> <xs:annotation> <xs:documentation>SDS1.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="1"> <xs:annotation> <xs:documentation>SDS2.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="2"> <xs:annotation> <xs:documentation>SDS3.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="3"> <xs:annotation> <xs:documentation>SDS4.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="4"></pre>																		

```

<xs:annotation>
  <xs:documentation>SDS-TL.</xs:documentation>
</xs:annotation>
</xs:enumeration>
<xs:enumeration value="5">
  <xs:annotation>
    <xs:documentation>Status.</xs:documentation>
  </xs:annotation>
</xs:enumeration>
</xs:restriction>
</xs:simpleType>

```

Simple Type typeReport

Namespace	DR-GW								
Annotations									
Diagram	<p>Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...</p>								
Type	restriction of xs:normalizedString								
Facets	<table> <tr> <td>enumeration</td> <td>none</td> </tr> <tr> <td>enumeration</td> <td>delivery</td> </tr> <tr> <td>enumeration</td> <td>consume</td> </tr> <tr> <td>enumeration</td> <td>both</td> </tr> </table>	enumeration	none	enumeration	delivery	enumeration	consume	enumeration	both
enumeration	none								
enumeration	delivery								
enumeration	consume								
enumeration	both								
Used by	Element typeSds/report								
Source	<pre> <xs:simpleType name="typeReport"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="none"/> <xs:enumeration value="delivery"/> <xs:enumeration value="consume"/> <xs:enumeration value="both"/> </xs:restriction> </xs:simpleType> </pre>								

Simple Type typeOrganisationBlockIdSimple

Namespace	DR-GW
Annotations	Organisation block send as simple normalized string. The pattern is: id1-id2-id3-id4-id5-id6
Diagram	<p>Organisation block send as simple normalized string. The pattern is: id1-id2-id3-id4-id5-id6</p> <p>Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...</p>
Type	restriction of xs:normalizedString
Facets	<p>pattern</p> <pre> ((([0-9] [1-9]\d{0,3} [1-5]\d{4} 6[0-4]\d{3} 65[0-4]\d{2} 655[0-2]\d 6553[0-5])-){0,5}([0-9] [1-9]\d{0,3} [1-5]\d{4} 6[0-4]\d{3} 65[0-4]\d{2} 655[0-2]\d 6553[0-5]) </pre>
Used by	Element typeOrganisationBlockId/orgblockIdSimple
Source	<pre> <xs:simpleType name="typeOrganisationBlockIdSimple"> <xs:annotation> <xs:documentation>Organisation block send as simple normalized string. The pattern is: id1-id2- id3-id4-id5-id6</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:pattern value="((([0-9] [1-9]\d{0,3} [1-5]\d{4} 6[0-4]\d{3} 65[0-4]\d{2} 655[0-2]\d 6553[0-5])-){0,5}([0-9] [1-9]\d{0,3} [1-5]\d{4} 6[0-4]\d{3} 65[0-4]\d{2} 655[0-2]\d 6553[0-5])" /> </xs:restriction> </xs:simpleType> </pre>

```

</xs:restriction>
</xs:simpleType>

```

Simple Type typeGroupTrackingMask

Namespace	DR-GW
Annotations	Bit mask of one or more typeGroupTrackingMaskValues using bitwise OR.
Diagram	<p>The diagram illustrates the definition of the typeGroupTrackingMask simple type. It shows a box labeled "typeGroupTrackingMask" connected by a line with a hollow circle to a box labeled "xs:unsignedShort". A callout box below "typeGroupTrackingMask" states: "Bit mask of one or more typeGroupTrackingMaskValues using bitwise OR.". Another callout box below "xs:unsignedShort" states: "Built-in derived type. The unsignedShort datatype is derived from unsignedInt by setting the value of maxInclusive to...".</p>
Type	xs:unsignedShort
Used by	Elements typeGroupTrack/mask, typeGroupTrackSubscriptionEvent/mask
Source	<pre> <xs:simpleType name="typeGroupTrackingMask"> <xs:annotation> <xs:documentation>Bit mask of one or more typeGroupTrackingMaskValues using bitwise OR.</xs:documentation> </xs:annotation> <xs:restriction base="xs:unsignedShort" /> </xs:simpleType> </pre>

Simple Type typeMembershipType

Namespace	DR-GW						
Annotations	Specifies a group - radio subscriber membership type.						
Diagram	<p>The diagram illustrates the definition of the typeMembershipType simple type. It shows a box labeled "typeMembershipType" connected by a line with a hollow circle to a box labeled "xs:normalizedString". A callout box below "typeMembershipType" states: "Specifies a group - radio subscriber membership type.". Another callout box below "xs:normalizedString" states: "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...".</p>						
Type	restriction of xs:normalizedString						
Facets	<table> <tr> <td>enumeration</td> <td>unknown</td> </tr> <tr> <td>enumeration</td> <td>permanent</td> </tr> <tr> <td>enumeration</td> <td>visiting</td> </tr> </table>	enumeration	unknown	enumeration	permanent	enumeration	visiting
enumeration	unknown						
enumeration	permanent						
enumeration	visiting						
Used by	Element typeGroupAddRadioMember/membership						
Source	<pre> <xs:simpleType name="typeMembershipType"> <xs:annotation> <xs:documentation>Specifies a group - radio subscriber membership type.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString" > <xs:enumeration value="unknown" /> <xs:enumeration value="permanent" /> <xs:enumeration value="visiting" /> </xs:restriction> </xs:simpleType> </pre>						

Simple Type typeGroupSelectionLevel

Namespace	DR-GW						
Annotations	Covers tcsScanningPriority_t of the TCS-API.						
Diagram	<p>The diagram illustrates the definition of the typeGroupSelectionLevel simple type. It shows a box labeled "typeGroupSelectionLevel" connected by a line with a hollow circle to a box labeled "xs:normalizedString". A callout box below "typeGroupSelectionLevel" states: "Covers tcsScanningPriority_t of the TCS-API.". Another callout box below "xs:normalizedString" states: "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...".</p>						
Type	restriction of xs:normalizedString						
Facets	<table> <tr> <td>enumeration</td> <td>notScanned</td> </tr> <tr> <td>enumeration</td> <td>low</td> </tr> <tr> <td>enumeration</td> <td>normal</td> </tr> </table>	enumeration	notScanned	enumeration	low	enumeration	normal
enumeration	notScanned						
enumeration	low						
enumeration	normal						

	enumeration	selected
	enumeration	high
	enumeration	background
Used by	Element	typeRadioGroupSelection/level
Source	<pre><xs:simpleType name="typeGroupSelectionLevel"> <xs:annotation> <xs:documentation>Covers tcsScanningPriority_t of the TCS-API.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="notScanned"/> <xs:enumeration value="low"/> <xs:enumeration value="normal"/> <xs:enumeration value="selected"/> <xs:enumeration value="high"/> <xs:enumeration value="background"/> </xs:restriction> </xs:simpleType></pre>	

Simple Type typeSystemElementState

Namespace	DR-GW										
Annotations	Specifies connection, server or unit state.										
Diagram	<pre> classDiagram typeSystemElementState < -- xs:normalizedString </pre> <p>The diagram shows a UML class named "typeSystemElementState" with a generalization arrow pointing to the built-in datatype "xs:normalizedString". A callout box below "typeSystemElementState" states "Specifies connection, server or unit state.". A callout box below "xs:normalizedString" states "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...".</p>										
Type	restriction of xs:normalizedString										
Facets	<table border="1"> <tr> <td>enumeration</td> <td>unknown</td> <td>Unknown state.</td> </tr> <tr> <td>enumeration</td> <td>ok</td> <td>Connection or server is working.</td> </tr> <tr> <td>enumeration</td> <td>n_Ok</td> <td>Connection or server is not working.</td> </tr> </table>		enumeration	unknown	Unknown state.	enumeration	ok	Connection or server is working.	enumeration	n_Ok	Connection or server is not working.
enumeration	unknown	Unknown state.									
enumeration	ok	Connection or server is working.									
enumeration	n_Ok	Connection or server is not working.									
Used by	Elements	typeSystemTetraStatesEvent/cddconnectionState, typeSystemTetraStatesEvent/cddserverState, typeSystemTetraStatesEvent/dxtState, typeSystemTetraStatesEvent/tcsState									
Source	<pre><xs:simpleType name="typeSystemElementState"> <xs:annotation> <xs:documentation>Specifies connection, server or unit state.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="unknown"> <xs:annotation> <xs:documentation>Unknown state.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="ok"> <xs:annotation> <xs:documentation>Connection or server is working.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="n_Ok"> <xs:annotation> <xs:documentation>Connection or server is not working.</xs:documentation> </xs:annotation> </xs:enumeration> </xs:restriction> </xs:simpleType></pre>										

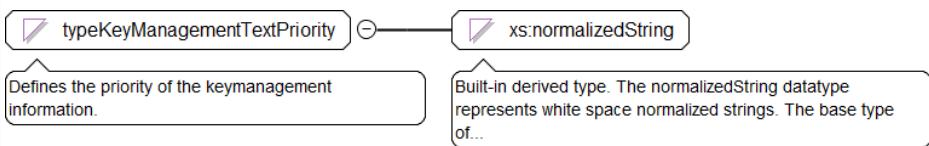
Simple Type typeActionRequest

Namespace	DR-GW	
Annotations	All possible call actions.	
Diagram	<pre> classDiagram typeActionRequest < -- xs:normalizedString </pre> <p>The diagram shows a UML class named "typeActionRequest" with a generalization arrow pointing to the built-in datatype "xs:normalizedString". A callout box below "typeActionRequest" states "All possible call actions.". A callout box below "xs:normalizedString" states "Built-in derived type. The normalizedString datatype represents white space normalized strings. The base type of...".</p>	

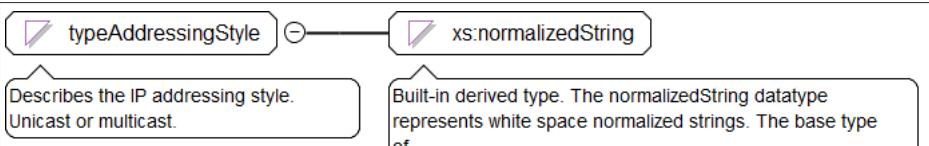
Type	restriction of xs:normalizedString	
Facets	enumeration	setup This method is used to initiate a new call setup. For a call setup to be successful it is required that the resources have been reserved prior this method call.
	enumeration	connect This method is used to connect an incoming call.
	enumeration	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.
Used by	Element	typeCallRequest/action
Source	<pre> <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> </pre>	

Simple Type typeKeyManagementTextPriority

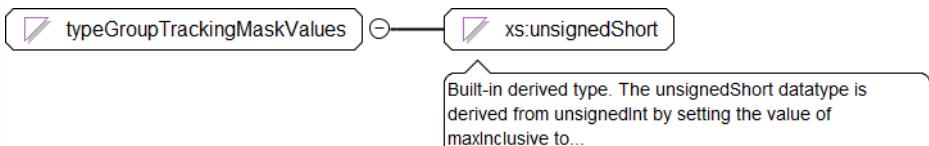
Namespace	DR-GW
Annotations	Defines the priority of the keymanagement information.

Diagram					
Type	restriction of xs:normalizedString				
Facets	<table> <tr> <td>enumeration</td> <td>normal</td> </tr> <tr> <td>enumeration</td> <td>high</td> </tr> </table>	enumeration	normal	enumeration	high
enumeration	normal				
enumeration	high				
Source	<pre> <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> </pre>				

Simple Type typeAddressingStyle

Namespace	DR-GW				
Annotations	Describes the IP addressing style. Unicast or multicast.				
Diagram					
Type	restriction of xs:normalizedString				
Facets	<table> <tr> <td>enumeration</td> <td>ucast</td> </tr> <tr> <td>enumeration</td> <td>mcast</td> </tr> </table>	enumeration	ucast	enumeration	mcast
enumeration	ucast				
enumeration	mcast				
Source	<pre> <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> </pre>				

Simple Type typeGroupTrackingMaskValues

Namespace	DR-GW																					
Annotations																						
Diagram																						
Type	restriction of xs:unsignedShort																					
Facets	<table> <tr> <td>enumeration</td> <td>0</td> <td>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_BASIC_C</td> </tr> <tr> <td>enumeration</td> <td>1</td> <td>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_RS_ADD_REMOVE_C</td> </tr> <tr> <td>enumeration</td> <td>2</td> <td>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_G4WIF_ADD_REMOVE_C</td> </tr> <tr> <td>enumeration</td> <td>4</td> <td>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_WSUSER_ADD_REMOVE_C</td> </tr> <tr> <td>enumeration</td> <td>8</td> <td>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_CBR_REMOVE_C</td> </tr> <tr> <td>enumeration</td> <td>16</td> <td>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_GROUP_ADD_REMOVE_C</td> </tr> <tr> <td>enumeration</td> <td>65535</td> <td>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_ALL_C</td> </tr> </table>	enumeration	0	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_BASIC_C	enumeration	1	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_RS_ADD_REMOVE_C	enumeration	2	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_G4WIF_ADD_REMOVE_C	enumeration	4	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_WSUSER_ADD_REMOVE_C	enumeration	8	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_CBR_REMOVE_C	enumeration	16	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_GROUP_ADD_REMOVE_C	enumeration	65535	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_ALL_C
enumeration	0	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_BASIC_C																				
enumeration	1	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_RS_ADD_REMOVE_C																				
enumeration	2	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_G4WIF_ADD_REMOVE_C																				
enumeration	4	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_WSUSER_ADD_REMOVE_C																				
enumeration	8	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_CBR_REMOVE_C																				
enumeration	16	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_GROUP_ADD_REMOVE_C																				
enumeration	65535	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_ALL_C																				
Source	<pre> <xs:simpleType name="typeGroupTrackingMaskValues"> </pre>																					

```

<xs:annotation>
  <xs:documentation></xs:documentation>
</xs:annotation>
<xs:restriction base="xs:unsignedShort">
  <xs:enumeration value="0">
    <xs:annotation>
      <xs:documentation>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_BASIC_C</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="1">
    <xs:annotation>
      <xs:documentation>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_RS_ADD_REMOVE_C</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="2">
    <xs:annotation>
      <xs:documentation>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_G4WIF_ADD_REMOVE_C</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="4">
    <xs:annotation>
      <xs:documentation>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_WSUSER_ADD_REMOVE_C</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="8">
    <xs:annotation>
      <xs:documentation>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_CBR_REMOVE_C</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="16">
    <xs:annotation>
      <xs:documentation>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_GROUP_ADD_REMOVE_C</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="65535">
    <xs:annotation>
      <xs:documentation>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_ALL_C</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
</xs:restriction>
</xs:simpleType>

```

Attribute(s)

Attribute drgw / @version

Namespace	DR-GW	
Properties	use:	required
	fixed:	2.0
Used by	Element	drgw
Source	<xs:attribute name="version" fixed="2.0" use="required"/>	