

文件比较1

统计日期：2008-2-25 1:51:36

模式：全部显示

左侧文件：D:\ttcn3_sources_updated\AtsBCast_TestControl.ttcn

右侧文件：D:\ttcn3_sources\AtsBCast_TestControl.ttcn

```

1/**
2 *
3 * @author
4 *   ETSI CTI BCAST CON Project
5 * @version
6 *   $Id$
7 * @desc
8 *   This module specifies a fairly
static test execution control. Via
9 *   module parameters for test area
or specific test case identifiers
10 *   execution can be controlled. The
BCAST test suite can also be
11 *   compiled without this module. In
that case test case execution must
12 *   be implemented via the TCI-TM
interface.
13 */
14module AtsBCast_TestControl {
15    import from
AtsBCast_ModuleParameters all;
16    import from
AtsBCast_ContentProtectionTests all;
17    import from
AtsBCast_FileAndStreamDistributionTests
all;
18    import from
AtsBCast_ServiceGuideTests all;
19    import from
AtsBCast_ServiceInteractionTests all;
20    import from
AtsBCast_ServiceProvisioningTests all;
21
22    // test case selection switches
23    control {
24        log ("Control part
started...");
25        if (PX_ALL_TCS) {
26            log ("Running
all test cases");
27        }
28        if (PX_ALL_TCS or
PX_ALL_SP_TCS) {
29            log ("Running
all service provisoring test cases");
30            execute
(TC_BCAST_PROV_CONF_101());

```

```

1/**
2 *
3 * @author
4 *   ETSI CTI BCAST CON Project
5 * @version
6 *   $Id$
7 * @desc
8 *   This module specifies a fairly
static test execution control. Via
9 *   module parameters for test area
or specific test case identifiers
10 *   execution can be controlled. The
BCAST test suite can also be
11 *   compiled without this module. In
that case test case execution must
12 *   be implemented via the TCI-TM
interface.
13 */
14module AtsBCast_TestControl {
15    import from
AtsBCast_ModuleParameters all;
16    import from
AtsBCast_ContentProtectionTests all;
17    import from
AtsBCast_FileAndStreamDistributionTests
all;
18    import from
AtsBCast_ServiceGuideTests all;
19    import from
AtsBCast_ServiceInteractionTests all;
20    import from
AtsBCast_ServiceProvisioningTests all;
21
22    // test case selection switches
23    control {
24        log ("Control part
started...");
25        if (PX_ALL_TCS) {
26            log ("Running
all test cases");
27        }
28        if (PX_ALL_TCS or
PX_ALL_SP_TCS) {
29            log ("Running
all service provisoring test cases");
30            execute
(TC_BCAST_PROV_CONF_101());

```

31	execute (TC_BCAST_PROV_CONF_102());	<>	31	execute (TC_BCAST_PROV_CONF_101());
32	}	=	32	}
33	if (PX_ALL_TCS or PX_ALL_SG_TCS) {		33	if (PX_ALL_TCS or PX_ALL_SG_TCS) {
34	log ("Running all service guide test cases");		34	log ("Running all service guide test cases");
35	execute (TC_BCAST_ESG_CONF_101());		35	execute (TC_BCAST_ESG_CONF_101());
36	execute (TC_BCAST_ESG_CONF_102());		36	execute (TC_BCAST_ESG_CONF_102());
37	execute (TC_BCAST_ESG_CONF_103());		37	execute (TC_BCAST_ESG_CONF_103());
38	execute (TC_BCAST_ESG_CONF_104());		38	execute (TC_BCAST_ESG_CONF_104());
39	execute (TC_BCAST_ESG_CONF_105());		39	execute (TC_BCAST_ESG_CONF_105());
40	execute (TC_BCAST_ESG_CONF_106());		40	execute (TC_BCAST_ESG_CONF_106());
41	execute (TC_BCAST_ESG_CONF_107());		41	execute (TC_BCAST_ESG_CONF_107());
42	execute (TC_BCAST_ESG_CONF_108());		42	execute (TC_BCAST_ESG_CONF_108());
43	}		43	}
44	if (PX_ALL_TCS or PX_ALL_FD_TCS) {		44	if (PX_ALL_TCS or PX_ALL_FD_TCS) {
45	log ("Running all file distribution test cases");		45	log ("Running all file distribution test cases");
46	execute (TC_BCAST_DIST_CONF_102());		46	execute (TC_BCAST_DIST_CONF_102());
47	}		47	}
48	if (PX_ALL_TCS or PX_ALL_SI_TCS) {		48	if (PX_ALL_TCS or PX_ALL_SI_TCS) {
49	log ("Running all service interaction test cases");		49	log ("Running all service interaction test cases");
50	execute (TC_BCAST_INTER_CONF_101());		50	execute (TC_BCAST_INTER_CONF_101());
51	execute (TC_BCAST_INTER_CONF_102());		51	execute (TC_BCAST_INTER_CONF_102());
52	execute (TC_BCAST_INTER_CONF_103());		52	execute (TC_BCAST_INTER_CONF_103());
53	}		53	}
54	if (PX_ALL_TCS or PX_ALL_CP_TCS) {		54	if (PX_ALL_TCS or PX_ALL_CP_TCS) {
55	log ("Running all content protection test cases");		55	log ("Running all content protection test cases");
56	execute (TC_BCAST_CONTPROT_CONF_101());		56	execute (TC_BCAST_CONTPROT_CONF_101());
57	execute (TC_BCAST_CONTPROT_CONF_102());		57	execute (TC_BCAST_CONTPROT_CONF_102());
58	execute (TC_BCAST_CONTPROT_CONF_103());		58	execute (TC_BCAST_CONTPROT_CONF_103());
59	}		59	}
60	}		60	}
61	// end module AtsBCast_TestControl		61	// end module AtsBCast_TestControl

文件比较1

统计日期：2008-2-25 1:52:00

模式：全部显示

左侧文件：D:\ttcn3_sources_updated\AtsBCast_ContentProtectionTests.ttcn

右侧文件：D:\ttcn3_sources\AtsBCast_ContentProtectionTests.ttcn

```
1/**
2 *
3 * @author
4 *   ETSI CTI BCAST CON Project
5 * @version
6 *   $Id$
7 * @desc
8 *   This module specifies pilot tests for BCast content protection.
9 */
10module AtsBCast_ContentProtectionTests {
11   import from LibBCast_Interface all;
12   import from AtsBCast_Main_Functions all;
13   import from LibBCast_ServiceGuide_Templates all;
14   import from AtsBCast_ServiceGuide_Functions all;
15   import from LibBCast_ServiceGuide_TypesAndValues all;
16   import from LibBCast_Common_TypesAndValues all;
17   import from LibCommon_BasicTypesAndValues all;
18   import from AtsBCast_ModuleParameters all;
19   import from LibBCast_ModuleParameters all;
20   import from AtsBCast_TestConfiguration_Functions all;
21   import from AtsBCast_TestSystem all;
22
23   group bcastConformanceTestCases {
24     group clientConformanceTestCases {
25
26       /**
27        *
28        * @desc
29        *   <p>Service protection with IPsec via
30broadcast channel (optional)</p>
31        *   <p>Test Case Description</p>
32        *   <p>with a terminal having received a
33service guide containing a service with purchase information on the
34Service Guide Delivery Channel and listening for an IPsec encrypted
35data stream on the broadcast channel</p>
36        *   <p>when the terminal subscribes to that
37service on the interactive channel using a Service Request/Response
38message exchange</p>
39        *   <p>then the terminal is able to receive,
40decrypt, and display the data stream correctly.</p>
41        *   <p>Specification Reference</p>
42        *   <p>[BCAST10-ServContProt] Section 6.8.1,
439.1.</p>
44        *   <p>Preconditions</p>
45        *   <p>Service guide cache of terminal is
46erased</p>
47        *   <p>Terminal is configured to listen to
48BCAST service guide announcements and delivery on the broadcast
49channel</p>
50        *   <p>Access point information for service
51guide entry point is configured in test tool</p>
```

```
=
1/**
2 *
3 * @author
4 *   ETSI CTI BCAST CON Project
5 * @version
6 *   $Id$
7 * @desc
8 *   This module specifies pilot tests for BCast content protection.
9 */
10module AtsBCast_ContentProtectionTests {
11   import from LibBCast_Interface all;
12   import from AtsBCast_Main_Functions all;
13   import from LibBCast_ServiceGuide_Templates all;
14   import from AtsBCast_ServiceGuide_Functions all;
15   import from LibBCast_ServiceGuide_TypesAndValues all;
16   import from LibBCast_Common_TypesAndValues all;
17   import from LibCommon_BasicTypesAndValues all;
18   import from AtsBCast_ModuleParameters all;
19   import from LibBCast_ModuleParameters all;
20   import from AtsBCast_TestConfiguration_Functions all;
21   import from AtsBCast_TestSystem all;
22
23   group bcastConformanceTestCases {
24     group clientConformanceTestCases {
25
26       /**
27        *
28        * @desc
29        *   <p>Service protection with IPsec via
30broadcast channel (optional)</p>
31        *   <p>Test Case Description</p>
32        *   <p>with a terminal having received a
33service guide containing a service with purchase information on the
34Service Guide Delivery Channel and listening for an IPsec encrypted
35data stream on the broadcast channel</p>
36        *   <p>when the terminal subscribes to that
37service on the interactive channel using a Service Request/Response
38message exchange</p>
39        *   <p>then the terminal is able to receive,
40decrypt, and display the data stream correctly.</p>
41        *   <p>Specification Reference</p>
42        *   <p>[BCAST10-ServContProt] Section 6.8.1,
439.1.</p>
44        *   <p>Preconditions</p>
45        *   <p>Service guide cache of terminal is
46erased</p>
47        *   <p>Terminal is configured to listen to
48BCAST service guide announcements and delivery on the broadcast
49channel</p>
50        *   <p>Access point information for service
51guide entry point is configured in test tool</p>
```

```
40      *      <p>This test cases uses the following SG
fragment instantiations found in the Appendix:
41      *      <ul>
42      *      <li>Service fragment with
Name="PayTvChannel "</li>
43      *      <li>Content fragment with
Name="Programme", and StartTime and EndTime elements indicating values
within the time of test</li>
44      *      <li>Schedule fragment for content
fragment with same values for startTime and endTime in the
presentationWindow element</li>
45      *      <li>Access fragment for schedule
fragment with KeyManagmentSystem, EncryptionType set to IPsec, and a
reference to the Session Description fragment </li>
46      *      <li>Session Description fragment
containing SDP for "Programme" and the service protection information
for IPsec</li>
47      *      <li>PurchaseItem fragment for
service fragment with Description "IPsec" element</li>
48      *      <li>PurchaseData fragment for
purchaseItem fragment with PriceInfo and Description "Discount"
elements</li>
49      *      <li>PurchaseChannel fragment for
purchaseItem fragment with PurchaseURL pointing to a subscription site</
li>
50      *      </ul>
51      *      </p>
52      *      <p>Note: All the fragments are associated
with the same Service fragment and are sent in the same service guide
delivery.</p>
53      *      <p>Test Procedure</p>
54      *      <p>
55      *      <ul>
56      *      <li>Set up the test tool to
produce BCAST service guide announcement and delivery using broadcast
channel.</li>
57      *      <li>Setup the test tool to stream
via IPsec encrypted audio and video for "Programme" as well as key
stream on broadcast channel</li>
58      *      <li>Setup the SEK delivery to the
terminal either via DRM 2.0 ROAP (DRM) or authentication and push
delivery (Smartcard) </li>
59      *      <li>Activate the BCAST application
of the terminal</li>
60      *      <li>Request from BCAST application
on terminal to get the service guide (optional)</li>
61      *      <li>Browse the SG in the terminal</
li>
62      *      <li>Setup the test tool to
authenticate and accept subscription request for the Service
"PayTvChannel" from the terminal via interactive channel</li>
63      *      <li>Purchase "TvChannel" on
terminal</li>
64      *      <li>Access "TvChannel" on
terminal</li>
65      *      </ul>
66      *      </p>
67      *      <p>Pass-Criteria</p>
```

```
40      *      <p>This test cases uses the following SG
fragment instantiations found in the Appendix:
41      *      <ul>
42      *      <li>Service fragment with
Name="PayTvChannel "</li>
43      *      <li>Content fragment with
Name="Programme", and StartTime and EndTime elements indicating values
within the time of test</li>
44      *      <li>Schedule fragment for content
fragment with same values for startTime and endTime in the
presentationWindow element</li>
45      *      <li>Access fragment for schedule
fragment with KeyManagmentSystem, EncryptionType set to IPsec, and a
reference to the Session Description fragment </li>
46      *      <li>Session Description fragment
containing SDP for "Programme" and the service protection information
for IPsec</li>
47      *      <li>PurchaseItem fragment for
service fragment with Description "IPsec" element</li>
48      *      <li>PurchaseData fragment for
purchaseItem fragment with PriceInfo and Description "Discount"
elements</li>
49      *      <li>PurchaseChannel fragment for
purchaseItem fragment with PurchaseURL pointing to a subscription site</
li>
50      *      </ul>
51      *      </p>
52      *      <p>Note: All the fragments are associated
with the same Service fragment and are sent in the same service guide
delivery.</p>
53      *      <p>Test Procedure</p>
54      *      <p>
55      *      <ul>
56      *      <li>Set up the test tool to
produce BCAST service guide announcement and delivery using broadcast
channel.</li>
57      *      <li>Setup the test tool to stream
via IPsec encrypted audio and video for "Programme" as well as key
stream on broadcast channel</li>
58      *      <li>Setup the SEK delivery to the
terminal either via DRM 2.0 ROAP (DRM) or authentication and push
delivery (Smartcard) </li>
59      *      <li>Activate the BCAST application
of the terminal</li>
60      *      <li>Request from BCAST application
on terminal to get the service guide (optional)</li>
61      *      <li>Browse the SG in the terminal</
li>
62      *      <li>Setup the test tool to
authenticate and accept subscription request for the Service
"PayTvChannel" from the terminal via interactive channel</li>
63      *      <li>Purchase "TvChannel" on
terminal</li>
64      *      <li>Access "TvChannel" on
terminal</li>
65      *      </ul>
66      *      </p>
67      *      <p>Pass-Criteria</p>
```

```

68      *      <p>The following should be visible to the
end user after the delivery of the SG
69      *      <ul>
70      *      <li>The service "PayTvChannel" is
displayed with "Programme" schedule from start to end time, available
as "Discount" purchase, and price information.</li>
71      *      <li>"Programme" can be viewed by
the end user.</li>
72      *      </ul>
73      *      </p>
74      *      <p>The following should be observable for
the test tool
75      *      <ul>
76      *      <li>The test tool receives HTTP
request from the end user to subscribe to service "TvChannel".</li>
77      *      </ul>
78      *      </p>
79      *      @verdict
80      *      pass in case the client pass the
conformance test.
81      *      @verdict
82      *      fail in case the client does not pass the
conformance test.
83      *      @verdict
84      *      inconc in case a guard timer expires.
85      */
86      testcase TC_BCAST_CONTPROT_CONF_101() runs on
BCastMainComponent system BCastPlatformComponent {
87          // init components and ports
88          f_createComponents();
89          f_cf_Broadcast_up();
90          f_cf_Interaction_up();
91          f_cf_UpperTester_up();
92
93          // preamble
94          f_startCommonUtPreamble();
95          // TODO: Omitted here is starting of
streaming server to broadcast the encrypted TV Channel program since
handling of SEK delivery is open issue
96
97          // test behavior
98          var ServiceGuideDeliveryUnit v_SGDU :=
{};
99          var UInt32 v_FragmentVersion := 1;
100
101          var UInt v_NTP_StartTime := 0;
102          var UInt v_NTP_EndTime := 0;
103          f_InitStartEndTime(v_NTP_StartTime,
v_NTP_EndTime);
104
105          var DateTime v_DateTime_StartTime :=
f_getDateTime(v_NTP_StartTime);
106          var DateTime v_DateTime_EndTime :=
f_getDateTime(v_NTP_EndTime);
107
108          var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
109              PX_SDP_VIDEO_PROG_1_REF_ID,
110              PX_SDP_SERVICE_PROTECTION_IPSEC
111          ));

```

```

68      *      <p>The following should be visible to the
end user after the delivery of the SG
69      *      <ul>
70      *      <li>The service "PayTvChannel" is
displayed with "Programme" schedule from start to end time, available
as "Discount" purchase, and price information.</li>
71      *      <li>"Programme" can be viewed by
the end user.</li>
72      *      </ul>
73      *      </p>
74      *      <p>The following should be observable for
the test tool
75      *      <ul>
76      *      <li>The test tool receives HTTP
request from the end user to subscribe to service "TvChannel".</li>
77      *      </ul>
78      *      </p>
79      *      @verdict
80      *      pass in case the client pass the
conformance test.
81      *      @verdict
82      *      fail in case the client does not pass the
conformance test.
83      *      @verdict
84      *      inconc in case a guard timer expires.
85      */
86      testcase TC_BCAST_CONTPROT_CONF_101() runs on
BCastMainComponent system BCastPlatformComponent {
87          // init components and ports
88          f_createComponents();
89          f_cf_Broadcast_up();
90          f_cf_Interaction_up();
91          f_cf_UpperTester_up();
92
93          // preamble
94          f_startCommonUtPreamble();
95          // TODO: Omitted here is starting of
streaming server to broadcast the encrypted TV Channel program since
handling of SEK delivery is open issue
96
97          // test behavior
98          var ServiceGuideDeliveryUnit v_SGDU :=
{};
99          var UInt32 v_FragmentVersion := 1;
100
101          var UInt v_NTP_StartTime := 0;
102          var UInt v_NTP_EndTime := 0;
103          f_InitStartEndTime(v_NTP_StartTime,
v_NTP_EndTime);
104
105          var DateTime v_DateTime_StartTime :=
f_getDateTime(v_NTP_StartTime);
106          var DateTime v_DateTime_EndTime :=
f_getDateTime(v_NTP_EndTime);
107
108          var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
109              PX_SDP_VIDEO_PROG_1_REF_ID,
110              PX_SDP_SERVICE_PROTECTION_IPSEC
111          ));

```

```
112
113 // TODO: Omitted here is setting of KMS
value - see open issues
114 var AccessTypeType v_AccessType :=
valueof(m_def_AccessType_bc_sdp(
115     f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
116     PX_SDP_VIDEO_PROG_1_REF_ID
117 ));
118
119 var PriceInfoType v_PriceInfo := valueof
(m_PriceInfo(
120     1,
121     PX_MONETARY_PRICE,
122     PX_CURRENCY
123 ));
124
125 var ServiceType v_Service := valueof
(m_def_Service(
126     PX_SGDU_SERVICE_ID,
127     v_FragmentVersion,
128     "PayTvChannel",
129 c_basicTv_ServiceType
130 ));
131
132 var ContentType v_Content := valueof
(m_def_Content(
133     PX_SGDU_CONTENT_ID_PROG_1,
134     v_FragmentVersion,
135     "Programme",
136     PX_SGDU_SERVICE_ID,
137     v_DateTime_StartTime,
138     v_DateTime_EndTime
139 ));
140
141 var ScheduleType v_Schedule := valueof
(m_def_Schedule(
142     PX_SGDU_SCHEDULE_ID_PROG_1,
143     v_FragmentVersion,
144     PX_SGDU_SERVICE_ID,
145     PX_SGDU_CONTENT_ID_PROG_1,
146     v_NTP_StartTime,
147     v_NTP_EndTime
148 ));
149
150 var AccessType v_Access := valueof
(m_def_Access(
151     PX_SGDU_ACCESS_ID_PROG_1,
152     v_FragmentVersion,
153     v_AccessType,
154     PX_SGDU_SERVICE_ID,
155     PX_SGDU_SCHEDULE_ID_PROG_1,
156     c_class_SG
157 ));
158
159 var PurchaseItemType v_PurchaseItem :=
valueof(m_def_PurchaseItem(
160     PX_SGDU_PURCHASE_ITEM_ID,
```

```
112
113 // TODO: Omitted here is setting of KMS
value - see open issues
114 var AccessTypeType v_AccessType :=
valueof(m_def_AccessType_bc_sdp(
115     f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
116     PX_SDP_VIDEO_PROG_1_REF_ID
117 ));
118
119 var PriceInfoType v_PriceInfo := valueof
(m_PriceInfo(
120     1,
121     PX_MONETARY_PRICE,
122     PX_CURRENCY
123 ));
124
125 var ServiceType v_Service := valueof
(m_def_Service(
126     PX_SGDU_SERVICE_ID,
127     v_FragmentVersion,
128     "PayTvChannel",
129 c_basicTv_ServiceType
130 ));
131
132 var ContentType v_Content := valueof
(m_def_Content(
133     PX_SGDU_CONTENT_ID_PROG_1,
134     v_FragmentVersion,
135     "Programme",
136     PX_SGDU_SERVICE_ID,
137     v_DateTime_StartTime,
138     v_DateTime_EndTime
139 ));
140
141 var ScheduleType v_Schedule := valueof
(m_def_Schedule(
142     PX_SGDU_SCHEDULE_ID_PROG_1,
143     v_FragmentVersion,
144     PX_SGDU_SERVICE_ID,
145     PX_SGDU_CONTENT_ID_PROG_1,
146     v_NTP_StartTime,
147     v_NTP_EndTime
148 ));
149
150 var AccessType v_Access := valueof
(m_def_Access(
151     PX_SGDU_ACCESS_ID_PROG_1,
152     v_FragmentVersion,
153     v_AccessType,
154     PX_SGDU_SERVICE_ID,
155     PX_SGDU_SCHEDULE_ID_PROG_1,
156     c_class_SG
157 ));
158
159 var PurchaseItemType v_PurchaseItem :=
valueof(m_def_PurchaseItem(
160     PX_SGDU_PURCHASE_ITEM_ID,
```

```
161         v_FragmentVersion,
162         PX_SGDU_PURCHASE_DATA_ID,
163         PX_SGDU_SERVICE_ID,
164         "PurchaseItem1"
165     ));
166     v_PurchaseItem.Descriptions := {{lang :
= omit, text_:= "IPsec"}}};
167
168     var PurchaseChannelType
v_PurchaseChannel := valueof(m_def_PurchaseChannel(
169         PX_SGDU_PURCHASE_CHANNEL_ID,
170         v_FragmentVersion,
171         "PurchaseChannel",
172         PX_PURCHASE_URL
173     ));
174
175     var PurchaseDataType v_PurchaseData :=
valueof(m_def_PurchaseData(
176         PX_SGDU_PURCHASE_DATA_ID,
177         v_FragmentVersion,
178         {{omit, "Discount"}}},
179         PX_SGDU_PURCHASE_ITEM_ID,
180         PX_SGDU_PURCHASE_CHANNEL_ID,
181         v_PriceInfo
182     ));
183
184     f_addServiceFragment(v_SGDU, v_Service);
185     f_addContentFragment(v_SGDU, v_Content);
186     f_addScheduleFragment(v_SGDU,
v_Schedule);
187
188     f_addAccessFragment(v_SGDU, v_Access);
189     f_addPurchaseItemFragment(v_SGDU,
v_PurchaseItem);
190
191     f_addPurchaseChannelFragment(v_SGDU,
v_PurchaseChannel);
192
193     f_addPurchaseDataFragment(v_SGDU,
v_PurchaseData);
194
195     f_addSdpFragment(v_SGDU,
v_FragmentVersion, v_Sdp);
196
197     f_main_ctrl_broadcastServiceGuide
(v_SGDU, omit);
198
199     f_main_ut_RunBCastApplication();
200     if (PX_GET_SG_DISPLAY_MANUALLY) {
201         f_main_ut_GetServiceGuide
(PX_SGDD_ID);
202     }
203     f_main_ut_CheckService("PayTvChannel");
204     f_main_ut_PurchaseService
("PayTvChannel");
205     f_main_ctrl_awaitingSubscription
("PayTvChannel"); // TODO: This function currently does not include any
checking and handling of authentication - see open issues
206     // TODO: Omitted here is generation of
proper acceptance of terminal purchase request - see open issues
207
208     f_main_ut_SelectService("PayTvChannel");
209     f_main_ut_UseService("PayTvChannel");
```

```
161         v_FragmentVersion,
162         PX_SGDU_PURCHASE_DATA_ID,
163         PX_SGDU_SERVICE_ID,
164         "PurchaseItem1"
165     ));
166     v_PurchaseItem.Descriptions := {{lang :
= omit, text_:= "IPsec"}}};
167
168     var PurchaseChannelType
v_PurchaseChannel := valueof(m_def_PurchaseChannel(
169         PX_SGDU_PURCHASE_CHANNEL_ID,
170         v_FragmentVersion,
171         "PurchaseChannel",
172         PX_PURCHASE_URL
173     ));
174
175     var PurchaseDataType v_PurchaseData :=
valueof(m_def_PurchaseData(
176         PX_SGDU_PURCHASE_DATA_ID,
177         v_FragmentVersion,
178         {{omit, "Discount"}}},
179         PX_SGDU_PURCHASE_ITEM_ID,
180         PX_SGDU_PURCHASE_CHANNEL_ID,
181         v_PriceInfo
182     ));
183
184     f_addServiceFragment(v_SGDU, v_Service);
185     f_addContentFragment(v_SGDU, v_Content);
186     f_addScheduleFragment(v_SGDU,
v_Schedule);
187
188     f_addAccessFragment(v_SGDU, v_Access);
189     f_addPurchaseItemFragment(v_SGDU,
v_PurchaseItem);
190
191     f_addPurchaseChannelFragment(v_SGDU,
v_PurchaseChannel);
192
193     f_addPurchaseDataFragment(v_SGDU,
v_PurchaseData);
194
195     f_addSdpFragment(v_SGDU,
v_FragmentVersion, v_Sdp);
196
197     f_main_ctrl_broadcastServiceGuide
(v_SGDU, omit);
198
199     f_main_ut_RunBCastApplication();
200     if (PX_GET_SG_DISPLAY_MANUALLY) {
201         f_main_ut_GetServiceGuide
(PX_SGDD_ID);
202     }
203     f_main_ut_CheckService("PayTvChannel");
204     f_main_ut_PurchaseService
("PayTvChannel");
205     f_main_ctrl_awaitingSubscription
("PayTvChannel"); // TODO: This function currently does not include any
checking and handling of authentication - see open issues
206     // TODO: Omitted here is generation of
proper acceptance of terminal purchase request - see open issues
207
208     f_main_ut_SelectService("PayTvChannel");
209     f_main_ut_UseService("PayTvChannel");
```

```
206         f_main_ut_CheckVideo("Programme");
207
208         // postamble
209         // TODO: Omitted here is stopping the
streaming server to broadcast the encrypted TV Channel program since
handling of SEK delivery is open issue
210         f_main_ut_PowerOff();
211
212         f_cf_Broadcast_down();
213         f_cf_Interaction_down();
214         f_cf_UpperTester_down();
215     } // end test case TC_BCAST_CONTPROT_CONF_101
216
217     /**
218     *
219     * @desc
220     * <p>Service protection with SRTP via
broadcast channel (optional)</p>
221     * <p>Test Case Description</p>
222     * <p>with a terminal having received a
service guide containing a service with purchase information on the
Service Guide Delivery Channel and listening for an SRTP encrypted data
stream on the broadcast channel</p>
223     * <p>when the terminal subscribes to that
service on the interactive channel Service Request/Response message
exchange</p>
224     * <p>then the terminal is able to receive,
decrypt, and display the data stream correctly</p>
225     * <p>Specification Reference</p>
226     * <p>[BCAST10-ServContProt] Section 6.8.1,
9.2.</p>
227     * <p>Preconditions</p>
228     * <p>Service guide cache of terminal is
erased</p>
229     * <p>Terminal is configured to listen to
BCAST service guide announcements and delivery on the broadcast
channel</p>
230     * <p>Access point information for service
guide entry point is configured in test tool</p>
231     * <p>This test cases uses the following SG
fragment instantiations found in the Appendix:
232     * <ul>
233     * <li>Service fragment with
Name="PayTvChannel "</li>
234     * <li>Content fragment with
Name="Programme", and StartTime and EndTime elements indicating values
within the time of test</li>
235     * <li>Schedule fragment for content
fragment with same values for startTime and endTime in the
presentationWindow element</li>
236     * <li>Access fragment for schedule
fragment with KeyManagmentSystem, EncryptionType set to SRTP, and a
reference to the Session Description fragment</li>
237     * <li>Session Description fragment
containing SDP for "Programme" and the service protection information
for SRTP</li>
238     * <li>PurchaseItem fragment for
service fragment with Description "SRTP" element</li>
```

```
206         f_main_ut_CheckVideo("Programme");
207
208         // postamble
209         // TODO: Omitted here is stopping the
streaming server to broadcast the encrypted TV Channel program since
handling of SEK delivery is open issue
210         f_main_ut_PowerOff();
211
212         f_cf_Broadcast_down();
213         f_cf_Interaction_down();
214         f_cf_UpperTester_down();
215     } // end test case TC_BCAST_INTER_CONF_102
216
217     /**
218     *
219     * @desc
220     * <p>Service protection with SRTP via
broadcast channel (optional)</p>
221     * <p>Test Case Description</p>
222     * <p>with a terminal having received a
service guide containing a service with purchase information on the
Service Guide Delivery Channel and listening for an SRTP encrypted data
stream on the broadcast channel</p>
223     * <p>when the terminal subscribes to that
service on the interactive channel Service Request/Response message
exchange</p>
224     * <p>then the terminal is able to receive,
decrypt, and display the data stream correctly</p>
225     * <p>Specification Reference</p>
226     * <p>[BCAST10-ServContProt] Section 6.8.1,
9.2.</p>
227     * <p>Preconditions</p>
228     * <p>Service guide cache of terminal is
erased</p>
229     * <p>Terminal is configured to listen to
BCAST service guide announcements and delivery on the broadcast
channel</p>
230     * <p>Access point information for service
guide entry point is configured in test tool</p>
231     * <p>This test cases uses the following SG
fragment instantiations found in the Appendix:
232     * <ul>
233     * <li>Service fragment with
Name="PayTvChannel "</li>
234     * <li>Content fragment with
Name="Programme", and StartTime and EndTime elements indicating values
within the time of test</li>
235     * <li>Schedule fragment for content
fragment with same values for startTime and endTime in the
presentationWindow element</li>
236     * <li>Access fragment for schedule
fragment with KeyManagmentSystem, EncryptionType set to SRTP, and a
reference to the Session Description fragment</li>
237     * <li>Session Description fragment
containing SDP for "Programme" and the service protection information
for SRTP</li>
238     * <li>PurchaseItem fragment for
service fragment with Description "SRTP" element</li>
```



```
239      *      <li>PurchaseData fragment for
purchaseItem fragment with PriceInfo and Description "Discount"
elements</li>
240      *      <li>PurchaseChannel fragment for
purchaseItem fragment with PurchaseURL pointing to a subscription site</
li>
241      *      </ul>
242      *      </p>
243      *      <p>Note: All the fragments are associated
with the same Service fragment and are sent in the same service guide
delivery.</p>
244      *      <p>Test Procedure</p>
245      *      <p>
246      *      <ul>
247      *      <li>Set up the test tool to
produce BCAST service guide announcement and delivery using broadcast
channel.</li>
248      *      <li>Setup the test tool to stream
via SRTP encrypted audio and video for "Programme" as well as key
stream on broadcast channel</li>
249      *      <li>Setup the SEK delivery to the
terminal either via DRM 2.0 ROAP (DRM) or authentication and push
delivery (Smartcard)</li>
250      *      <li>Activate the BCAST application
of the terminal</li>
251      *      <li>Request from BCAST application
on terminal to get the service guide (optional)</li>
252      *      <li>Browse the SG in the terminal</
li>
253      *      <li>Setup the test tool to
authenticate and accept subscription request for the Service
"PayTvChannel" from the terminal via interactive channel</li>
254      *      <li>Purchase "TvChannel" on
terminal</li>
255      *      <li>Access "TvChannel" on
terminal</li>
256      *      </ul>
257      *      </p>
258      *      <p>Pass-Criteria</p>
259      *      <p>The following should be visible to the
end user after the delivery of the SG
260      *      <ul>
261      *      <li>The service "PayTvChannel" is
displayed with "Programme" schedule from start to end time, available
as "Discount" purchase, and price information.</li>
262      *      <li>"Programme" can be viewed by
the end user.</li>
263      *      </ul>
264      *      </p>
265      *      <p>The following should be observable for
the test tool
266      *      <ul>
267      *      <li>The test tool receives HTTP
request from the end user to subscribe to service "TvChannel".</li>
268      *      </ul>
269      *      </p>
270      *      @verdict
271      *      pass in case the client pass the
conformance test.
```

```
239      *      <li>PurchaseData fragment for
purchaseItem fragment with PriceInfo and Description "Discount"
elements</li>
240      *      <li>PurchaseChannel fragment for
purchaseItem fragment with PurchaseURL pointing to a subscription site</
li>
241      *      </ul>
242      *      </p>
243      *      <p>Note: All the fragments are associated
with the same Service fragment and are sent in the same service guide
delivery.</p>
244      *      <p>Test Procedure</p>
245      *      <p>
246      *      <ul>
247      *      <li>Set up the test tool to
produce BCAST service guide announcement and delivery using broadcast
channel.</li>
248      *      <li>Setup the test tool to stream
via SRTP encrypted audio and video for "Programme" as well as key
stream on broadcast channel</li>
249      *      <li>Setup the SEK delivery to the
terminal either via DRM 2.0 ROAP (DRM) or authentication and push
delivery (Smartcard)</li>
250      *      <li>Activate the BCAST application
of the terminal</li>
251      *      <li>Request from BCAST application
on terminal to get the service guide (optional)</li>
252      *      <li>Browse the SG in the terminal</
li>
253      *      <li>Setup the test tool to
authenticate and accept subscription request for the Service
"PayTvChannel" from the terminal via interactive channel</li>
254      *      <li>Purchase "TvChannel" on
terminal</li>
255      *      <li>Access "TvChannel" on
terminal</li>
256      *      </ul>
257      *      </p>
258      *      <p>Pass-Criteria</p>
259      *      <p>The following should be visible to the
end user after the delivery of the SG
260      *      <ul>
261      *      <li>The service "PayTvChannel" is
displayed with "Programme" schedule from start to end time, available
as "Discount" purchase, and price information.</li>
262      *      <li>"Programme" can be viewed by
the end user.</li>
263      *      </ul>
264      *      </p>
265      *      <p>The following should be observable for
the test tool
266      *      <ul>
267      *      <li>The test tool receives HTTP
request from the end user to subscribe to service "TvChannel".</li>
268      *      </ul>
269      *      </p>
270      *      @verdict
271      *      pass in case the client pass the
conformance test.
```

```

272         * @verdict
273         *   fail in case the client does not pass the
conformance test.
274         * @verdict
275         *   inconc in case a guard timer expires.
276         */
277         testcase TC_BCAST_CONTPROT_CONF_102() runs on
BCastMainComponent system BCastPlatformComponent {
278             // init components and ports
279             f_createComponents();
280             f_cf_Broadcast_up();
281             f_cf_Interaction_up();
282             f_cf_UpperTester_up();
283
284             // preamble
285             f_startCommonUtPreamble();
286             // TODO: Omitted here is starting of
streaming server to broadcast the encrypted TV Channel program since
handling of SEK delivery is open issue
287
288             // test behavior
289             var ServiceGuideDeliveryUnit v_SGDU :=
{
290                 var UInt32 v_FragmentVersion := 1;
291
292                 var UInt v_NTP_StartTime := 0;
293                 var UInt v_NTP_EndTime := 0;
294                 f_InitStartEndTime(v_NTP_StartTime,
v_NTP_EndTime);
295
296                 var DateTime v_DateTime_StartTime :=
f_getDateTime(v_NTP_StartTime);
297                 var DateTime v_DateTime_EndTime :=
f_getDateTime(v_NTP_EndTime);
298
299                 var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
300                     PX_SDP_VIDEO_PROG_1_REF_ID,
301                     PX_SDP_SERVICE_PROTECTION_SRTTP
302                 ));
303
304                 var AccessTypeType v_AccessType :=
valueof(m_def_AccessType_bc_sdp(
305                     f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
306                     PX_SDP_VIDEO_PROG_1_REF_ID
307                 ));
308
309                 var PriceInfoType v_PriceInfo := valueof
(m_PriceInfo(
310                     1,
311                     PX_MONETARY_PRICE,
312                     PX_CURRENCY
313                 ));
314
315                 var ServiceType v_Service := valueof
(m_def_Service(
316                     PX_SGDU_SERVICE_ID,
317                     v_FragmentVersion,

```

```

272         * @verdict
273         *   fail in case the client does not pass the
conformance test.
274         * @verdict
275         *   inconc in case a guard timer expires.
276         */
277         testcase TC_BCAST_CONTPROT_CONF_102() runs on
BCastMainComponent system BCastPlatformComponent {
278             // init components and ports
279             f_createComponents();
280             f_cf_Broadcast_up();
281             f_cf_Interaction_up();
282             f_cf_UpperTester_up();
283
284             // preamble
285             f_startCommonUtPreamble();
286             // TODO: Omitted here is starting of
streaming server to broadcast the encrypted TV Channel program since
handling of SEK delivery is open issue
287
288             // test behavior
289             var ServiceGuideDeliveryUnit v_SGDU :=
{
290                 var UInt32 v_FragmentVersion := 1;
291
292                 var UInt v_NTP_StartTime := 0;
293                 var UInt v_NTP_EndTime := 0;
294                 f_InitStartEndTime(v_NTP_StartTime,
v_NTP_EndTime);
295
296                 var DateTime v_DateTime_StartTime :=
f_getDateTime(v_NTP_StartTime);
297                 var DateTime v_DateTime_EndTime :=
f_getDateTime(v_NTP_EndTime);
298
299                 var SdpFragment v_Sdp := valueof
(m_def_SdpFragment(
300                     PX_SDP_VIDEO_PROG_1_REF_ID,
301                     PX_SDP_SERVICE_PROTECTION_SRTTP
302                 ));
303
304                 var AccessTypeType v_AccessType :=
valueof(m_def_AccessType_bc_sdp(
305                     f_getAccessType
(PX_BROADCAST_NETWORK_BEARER),
306                     PX_SDP_VIDEO_PROG_1_REF_ID
307                 ));
308
309                 var PriceInfoType v_PriceInfo := valueof
(m_PriceInfo(
310                     1,
311                     PX_MONETARY_PRICE,
312                     PX_CURRENCY
313                 ));
314
315                 var ServiceType v_Service := valueof
(m_def_Service(
316                     PX_SGDU_SERVICE_ID,
317                     v_FragmentVersion,

```

```

318         "PayTvChannel",
319         c_basicTv_ServiceType
320     ));
321
322     var ContentType v_Content := valueof
323     (m_def_Content(
324         PX_SGDU_CONTENT_ID_PROG_1,
325         v_FragmentVersion,
326         "Programme",
327         PX_SGDU_SERVICE_ID,
328         v_DateTime_StartTime,
329         v_DateTime_EndTime
330     ));
331
332     var ScheduleType v_Schedule := valueof
333     (m_def_Schedule(
334         PX_SGDU_SCHEDULE_ID_PROG_1,
335         v_FragmentVersion,
336         PX_SGDU_SERVICE_ID,
337         PX_SGDU_CONTENT_ID_PROG_1,
338         v_NTP_StartTime,
339         v_NTP_EndTime
340     ));
341
342     // TODO: Omitted here is setting of KMS
343     value - see open issues
344
345     var AccessType v_Access := valueof
346     (m_def_Access(
347         PX_SGDU_ACCESS_ID_PROG_1,
348         v_FragmentVersion,
349         v_AccessType,
350         PX_SGDU_SERVICE_ID,
351         PX_SGDU_SCHEDULE_ID_PROG_1,
352         c_class_SG
353     ));
354
355     var PurchaseItemType v_PurchaseItem :=
356     valueof(m_def_PurchaseItem(
357         PX_SGDU_PURCHASE_ITEM_ID,
358         v_FragmentVersion,
359         PX_SGDU_PURCHASE_DATA_ID,
360         PX_SGDU_SERVICE_ID,
361         "PurchaseItem1"
362     ));
363
364     v_PurchaseItem.Descriptions := {{lang :
365     = omit, text_:= "SRTP"}}};
366
367     var PurchaseChannelType
368     v_PurchaseChannel := valueof(m_def_PurchaseChannel(
369         PX_SGDU_PURCHASE_CHANNEL_ID,
370         v_FragmentVersion,
371         "PurchaseChannel",
372         PX_PURCHASE_URL
373     ));
374
375     var PurchaseDataType v_PurchaseData :=
376     valueof(m_def_PurchaseData(
377         PX_SGDU_PURCHASE_DATA_ID,
378         v_FragmentVersion,
379         {{omit, "Discount"}}},

```

```

318         "PayTvChannel",
319         c_basicTv_ServiceType
320     ));
321
322     var ContentType v_Content := valueof
323     (m_def_Content(
324         PX_SGDU_CONTENT_ID_PROG_1,
325         v_FragmentVersion,
326         "Programme",
327         PX_SGDU_SERVICE_ID,
328         v_DateTime_StartTime,
329         v_DateTime_EndTime
330     ));
331
332     var ScheduleType v_Schedule := valueof
333     (m_def_Schedule(
334         PX_SGDU_SCHEDULE_ID_PROG_1,
335         v_FragmentVersion,
336         PX_SGDU_SERVICE_ID,
337         PX_SGDU_CONTENT_ID_PROG_1,
338         v_NTP_StartTime,
339         v_NTP_EndTime
340     ));
341
342     // TODO: Omitted here is setting of KMS
343     value - see open issues
344
345     var AccessType v_Access := valueof
346     (m_def_Access(
347         PX_SGDU_ACCESS_ID_PROG_1,
348         v_FragmentVersion,
349         v_AccessType,
350         PX_SGDU_SERVICE_ID,
351         PX_SGDU_SCHEDULE_ID_PROG_1,
352         c_class_SG
353     ));
354
355     var PurchaseItemType v_PurchaseItem :=
356     valueof(m_def_PurchaseItem(
357         PX_SGDU_PURCHASE_ITEM_ID,
358         v_FragmentVersion,
359         PX_SGDU_PURCHASE_DATA_ID,
360         PX_SGDU_SERVICE_ID,
361         "PurchaseItem1"
362     ));
363
364     v_PurchaseItem.Descriptions := {{lang :
365     = omit, text_:= "SRTP"}}};
366
367     var PurchaseChannelType
368     v_PurchaseChannel := valueof(m_def_PurchaseChannel(
369         PX_SGDU_PURCHASE_CHANNEL_ID,
370         v_FragmentVersion,
371         "PurchaseChannel",
372         PX_PURCHASE_URL
373     ));
374
375     var PurchaseDataType v_PurchaseData :=
376     valueof(m_def_PurchaseData(
377         PX_SGDU_PURCHASE_DATA_ID,
378         v_FragmentVersion,
379         {{omit, "Discount"}}},

```

```
370          PX_SGDU_PURCHASE_ITEM_ID,  
371          PX_SGDU_PURCHASE_CHANNEL_ID,  
372          v_PriceInfo  
373      ));  
374  
375      f_addServiceFragment(v_SGDU, v_Service);  
376      f_addContentFragment(v_SGDU, v_Content);  
377      f_addScheduleFragment(v_SGDU,  
v_Schedule);  
378      f_addAccessFragment(v_SGDU, v_Access);  
379      f_addPurchaseItemFragment(v_SGDU,  
v_PurchaseItem);  
380      f_addPurchaseChannelFragment(v_SGDU,  
v_PurchaseChannel);  
381      f_addPurchaseDataFragment(v_SGDU,  
v_PurchaseData);  
382      f_addSdpFragment(v_SGDU,  
v_FragmentVersion, v_Sdp);  
383  
384      f_main_ctrl_broadcastServiceGuide  
(v_SGDU, omit);  
385  
386      f_main_ut_RunBCastApplication();  
387      if (PX_GET_SG_DISPLAY_MANUALLY) {  
388          f_main_ut_GetServiceGuide  
(PX_SGDD_ID);  
389      }  
390      f_main_ut_CheckService("PayTvChannel");  
391      f_main_ut_PurchaseService  
("PayTvChannel");  
392      f_main_ctrl_awaitingSubscription  
("PayTvChannel"); // TODO: This function currently does not include any  
checking and handling of authentication - see open issues  
393      // TODO: Omitted here is generation of  
proper acceptance of terminal purchase request - see open issues  
394  
395      f_main_ut_SelectService("PayTvChannel");  
396      f_main_ut_UseService("PayTvChannel");  
397      f_main_ut_CheckVideo("Programme");  
398  
399      // postamble  
400      // TODO: Omitted here is stopping the  
streaming server to broadcast the encrypted TV Channel program since  
handling of SEK delivery is open issue  
401      f_main_ut_PowerOff();  
402  
403      f_cf_Broadcast_down();  
404      f_cf_Interaction_down();  
405      f_cf_UpperTester_down();  
406      } // end test case TC_BCAST_CONTPROT_CONF_102  
407  
408      /**  
409      *  
410      * @desc  
411      * <p>Service protection with ISMACrypt via  
broadcast channel (optional)</p>  
412      * <p>Test Case Description</p>
```

```
370          PX_SGDU_PURCHASE_ITEM_ID,  
371          PX_SGDU_PURCHASE_CHANNEL_ID,  
372          v_PriceInfo  
373      ));  
374  
375      f_addServiceFragment(v_SGDU, v_Service);  
376      f_addContentFragment(v_SGDU, v_Content);  
377      f_addScheduleFragment(v_SGDU,  
v_Schedule);  
378      f_addAccessFragment(v_SGDU, v_Access);  
379      f_addPurchaseItemFragment(v_SGDU,  
v_PurchaseItem);  
380      f_addPurchaseChannelFragment(v_SGDU,  
v_PurchaseChannel);  
381      f_addPurchaseDataFragment(v_SGDU,  
v_PurchaseData);  
382      f_addSdpFragment(v_SGDU,  
v_FragmentVersion, v_Sdp);  
383  
384      f_main_ctrl_broadcastServiceGuide  
(v_SGDU, omit);  
385  
386      f_main_ut_RunBCastApplication();  
387      if (PX_GET_SG_DISPLAY_MANUALLY) {  
388          f_main_ut_GetServiceGuide  
(PX_SGDD_ID);  
389      }  
390      f_main_ut_CheckService("PayTvChannel");  
391      f_main_ut_PurchaseService  
("PayTvChannel");  
392      f_main_ctrl_awaitingSubscription  
("PayTvChannel"); // TODO: This function currently does not include any  
checking and handling of authentication - see open issues  
393      // TODO: Omitted here is generation of  
proper acceptance of terminal purchase request - see open issues  
394  
395      f_main_ut_SelectService("PayTvChannel");  
396      f_main_ut_UseService("PayTvChannel");  
397      f_main_ut_CheckVideo("Programme");  
398  
399      // postamble  
400      // TODO: Omitted here is stopping the  
streaming server to broadcast the encrypted TV Channel program since  
handling of SEK delivery is open issue  
401      f_main_ut_PowerOff();  
402  
403      f_cf_Broadcast_down();  
404      f_cf_Interaction_down();  
405      f_cf_UpperTester_down();  
406      } // end test case TC_BCAST_INTER_CONF_102  
407  
408      /**  
409      *  
410      * @desc  
411      * <p>Service protection with ISMACrypt via  
broadcast channel (optional)</p>  
412      * <p>Test Case Description</p>
```

```
413      *      <p>with a terminal having received a
service guide containing a service with purchase information on the
Service Guide Delivery Channel and listening for an ISMACrypt encrypted
data stream on the broadcast channel</p>
414      *      <p>when the terminal subscribes to that
service on the interactive channel Service Request/Response message
exchange</p>
415      *      <p>then the terminal is able to receive,
decrypt, and display the data stream correctly</p>
416      *      <p>Specification Reference</p>
417      *      <p>[BCAST10-ServContProt] Section 6.8.1,
9.3.</p>
418      *      <p>Preconditions</p>
419      *      <p>Service guide cache of terminal is
erased</p>
420      *      <p>Terminal is configured to listen to
BCAST service guide announcements and delivery on the broadcast
channel</p>
421      *      <p>Access point information for service
guide entry point is configured in test tool</p>
422      *      <p>This test cases uses the following SG
fragment instantiations found in the Appendix:
423      *      <ul>
424      *          <li>Service fragment with
Name="PayTvChannel "</li>
425      *          <li>Content fragment with
Name="Programme", and StartTime and EndTime elements indicating values
within the time of test</li>
426      *          <li>Schedule fragment for content
fragment with same values for startTime and endTime in the
presentationWindow element</li>
427      *          <li>Access fragment for schedule
fragment with KeyManagmentSystem, EncryptionType set to ISMACrypt, and
a reference to the Session Description fragment</li>
428      *          <li>Session Description fragment
containing SDP for "Programme" and the service protection information
for ISMACrypt</li>
429      *          <li>PurchaseItem fragment for
service fragment with Description "ISMACrypt" element</li>
430      *          <li>PurchaseData fragment for
purchaseItem fragment with PriceInfo and Description "Discount"
elements</li>
431      *          <li>PurchaseChannel fragment for
purchaseItem fragment with PurchaseURL pointing to a subscription site</li>
432      *      </ul>
433      *      </p>
434      *      <p>Note: All the fragments are associated
with the same Service fragment and are sent in the same service guide
delivery.</p>
435      *      <p>Test Procedure</p>
436      *      <p>
437      *      <ul>
438      *          <li>Set up the test tool to
produce BCAST service guide announcement and delivery using broadcast
channel.</li>
439      *          <li>Setup the test tool to stream
via ISMACrypt encrypted audio and video for "Programme" as well as key
stream on broadcast channel</li>
```

```
413      *      <p>with a terminal having received a
service guide containing a service with purchase information on the
Service Guide Delivery Channel and listening for an ISMACrypt encrypted
data stream on the broadcast channel</p>
414      *      <p>when the terminal subscribes to that
service on the interactive channel Service Request/Response message
exchange</p>
415      *      <p>then the terminal is able to receive,
decrypt, and display the data stream correctly</p>
416      *      <p>Specification Reference</p>
417      *      <p>[BCAST10-ServContProt] Section 6.8.1,
9.3.</p>
418      *      <p>Preconditions</p>
419      *      <p>Service guide cache of terminal is
erased</p>
420      *      <p>Terminal is configured to listen to
BCAST service guide announcements and delivery on the broadcast
channel</p>
421      *      <p>Access point information for service
guide entry point is configured in test tool</p>
422      *      <p>This test cases uses the following SG
fragment instantiations found in the Appendix:
423      *      <ul>
424      *          <li>Service fragment with
Name="PayTvChannel "</li>
425      *          <li>Content fragment with
Name="Programme", and StartTime and EndTime elements indicating values
within the time of test</li>
426      *          <li>Schedule fragment for content
fragment with same values for startTime and endTime in the
presentationWindow element</li>
427      *          <li>Access fragment for schedule
fragment with KeyManagmentSystem, EncryptionType set to ISMACrypt, and
a reference to the Session Description fragment</li>
428      *          <li>Session Description fragment
containing SDP for "Programme" and the service protection information
for ISMACrypt</li>
429      *          <li>PurchaseItem fragment for
service fragment with Description "ISMACrypt" element</li>
430      *          <li>PurchaseData fragment for
purchaseItem fragment with PriceInfo and Description "Discount"
elements</li>
431      *          <li>PurchaseChannel fragment for
purchaseItem fragment with PurchaseURL pointing to a subscription site</li>
432      *      </ul>
433      *      </p>
434      *      <p>Note: All the fragments are associated
with the same Service fragment and are sent in the same service guide
delivery.</p>
435      *      <p>Test Procedure</p>
436      *      <p>
437      *      <ul>
438      *          <li>Set up the test tool to
produce BCAST service guide announcement and delivery using broadcast
channel.</li>
439      *          <li>Setup the test tool to stream
via ISMACrypt encrypted audio and video for "Programme" as well as key
stream on broadcast channel</li>
```

```
440      *      <li>Setup the SEK delivery to the
terminal either via DRM 2.0 ROAP (DRM) or authentication and push
delivery (Smartcard)</li>
441      *      <li>Activate the BCAST application
of the terminal</li>
442      *      <li>Request from BCAST application
on terminal to get the service guide (optional)</li>
443      *      <li>Browse the SG in the terminal</li>
444      *      <li>Setup the test tool to
authenticate and accept subscription request for the Service
"PayTvChannel" from the terminal via interactive channel</li>
445      *      <li>Purchase "TvChannel" on
terminal</li>
446      *      <li>Access "TvChannel" on
terminal</li>
447      *      </ul>
448      *      </p>
449      *      <p>Pass-Criteria</p>
450      *      <p>The following should be visible to the
end user after the delivery of the SG
451      *      <ul>
452      *      <li>The service "PayTvChannel" is
displayed with "Programme" schedule from start to end time, available
as "Discount" purchase, and price information.</li>
453      *      <li>"Programme" can be viewed by
the end user.</li>
454      *      </ul>
455      *      </p>
456      *      <p>The following should be observable for
the test tool
457      *      <ul>
458      *      <li>The test tool receives HTTP
request from the end user to subscribe to service "TvChannel".</li>
459      *      </ul>
460      *      </p>
461      *      @verdict
462      *      pass in case the client pass the
conformance test.
463      *      @verdict
464      *      fail in case the client does not pass the
conformance test.
465      *      @verdict
466      *      inconc in case a guard timer expires.
467      */
468      testcase TC_BCAST_CONTPROT_CONF_103() runs on
BCastMainComponent system BCastPlatformComponent {
469          // init components and ports
470          f_createComponents();
471          f_cf_Broadcast_up();
472          f_cf_Interaction_up();
473          f_cf_UpperTester_up();
474
475          // preamble
476          f_startCommonUtPreamble();
477          // TODO: Omitted here is starting of
streaming server to broadcast the encrypted TV Channel program since
handling of SEK delivery is open issue
478
479          // test behavior
```

```
440      *      <li>Setup the SEK delivery to the
terminal either via DRM 2.0 ROAP (DRM) or authentication and push
delivery (Smartcard)</li>
441      *      <li>Activate the BCAST application
of the terminal</li>
442      *      <li>Request from BCAST application
on terminal to get the service guide (optional)</li>
443      *      <li>Browse the SG in the terminal</li>
444      *      <li>Setup the test tool to
authenticate and accept subscription request for the Service
"PayTvChannel" from the terminal via interactive channel</li>
445      *      <li>Purchase "TvChannel" on
terminal</li>
446      *      <li>Access "TvChannel" on
terminal</li>
447      *      </ul>
448      *      </p>
449      *      <p>Pass-Criteria</p>
450      *      <p>The following should be visible to the
end user after the delivery of the SG
451      *      <ul>
452      *      <li>The service "PayTvChannel" is
displayed with "Programme" schedule from start to end time, available
as "Discount" purchase, and price information.</li>
453      *      <li>"Programme" can be viewed by
the end user.</li>
454      *      </ul>
455      *      </p>
456      *      <p>The following should be observable for
the test tool
457      *      <ul>
458      *      <li>The test tool receives HTTP
request from the end user to subscribe to service "TvChannel".</li>
459      *      </ul>
460      *      </p>
461      *      @verdict
462      *      pass in case the client pass the
conformance test.
463      *      @verdict
464      *      fail in case the client does not pass the
conformance test.
465      *      @verdict
466      *      inconc in case a guard timer expires.
467      */
468      testcase TC_BCAST_CONTPROT_CONF_103() runs on
BCastMainComponent system BCastPlatformComponent {
469          // init components and ports
470          f_createComponents();
471          f_cf_Broadcast_up();
472          f_cf_Interaction_up();
473          f_cf_UpperTester_up();
474
475          // preamble
476          f_startCommonUtPreamble();
477          // TODO: Omitted here is starting of
streaming server to broadcast the encrypted TV Channel program since
handling of SEK delivery is open issue
478
479          // test behavior
```

```
480|                                     var ServiceGuideDeliveryUnit v_SGDU :=  
481|                                     {};  
482|                                     var UInt32 v_FragmentVersion := 1;  
483|                                     var UInt v_NTP_StartTime := 0;  
484|                                     var UInt v_NTP_EndTime := 0;  
485|                                     f_InitStartEndTime(v_NTP_StartTime,  
v_NTP_EndTime);  
486|                                     var DateTime v_DateTime_StartTime :=  
487|                                     f_getDateTime(v_NTP_StartTime);  
488|                                     var DateTime v_DateTime_EndTime :=  
489|                                     f_getDateTime(v_NTP_EndTime);  
490|                                     var SdpFragment v_Sdp := valueof  
491|                                     (m_def_SdpFragment(  
492|                                     PX_SDP_VIDEO_PROG_1_REF_ID,  
493|                                     PX_SDP_SERVICE_PROTECTION_ISMA  
494|                                     ));  
495|                                     var AccessTypeType v_AccessType :=  
496|                                     valueof(m_def_AccessType_bc_sdp(  
497|                                     f_getAccessType  
498|                                     (PX_BROADCAST_NETWORK_BEARER),  
499|                                     PX_SDP_VIDEO_PROG_1_REF_ID  
500|                                     ));  
501|                                     var PriceInfoType v_PriceInfo := valueof  
502|                                     (m_PriceInfo(  
503|                                     1,  
504|                                     PX_MONETARY_PRICE,  
505|                                     PX_CURRENCY  
506|                                     ));  
507|                                     var ServiceType v_Service := valueof  
508|                                     (m_def_Service(  
509|                                     PX_SGDU_SERVICE_ID,  
510|                                     v_FragmentVersion,  
511|                                     "PayTvChannel",  
512|                                     c_basicTv_ServiceType  
513|                                     ));  
514|                                     var ContentType v_Content := valueof  
515|                                     (m_def_Content(  
516|                                     PX_SGDU_CONTENT_ID_PROG_1,  
517|                                     v_FragmentVersion,  
518|                                     "Programme",  
519|                                     PX_SGDU_SERVICE_ID,  
520|                                     v_DateTime_StartTime,  
521|                                     v_DateTime_EndTime  
522|                                     ));  
523|                                     var ScheduleType v_Schedule := valueof  
524|                                     (m_def_Schedule(  
525|                                     PX_SGDU_SCHEDULE_ID_PROG_1,  
526|                                     v_FragmentVersion,  
527|                                     PX_SGDU_SERVICE_ID,  
PX_SGDU_CONTENT_ID_PROG_1,  
v_NTP_StartTime,
```

```
480|                                     var ServiceGuideDeliveryUnit v_SGDU :=  
481|                                     {};  
482|                                     var UInt32 v_FragmentVersion := 1;  
483|                                     var UInt v_NTP_StartTime := 0;  
484|                                     var UInt v_NTP_EndTime := 0;  
485|                                     f_InitStartEndTime(v_NTP_StartTime,  
v_NTP_EndTime);  
486|                                     var DateTime v_DateTime_StartTime :=  
487|                                     f_getDateTime(v_NTP_StartTime);  
488|                                     var DateTime v_DateTime_EndTime :=  
489|                                     f_getDateTime(v_NTP_EndTime);  
490|                                     var SdpFragment v_Sdp := valueof  
491|                                     (m_def_SdpFragment(  
492|                                     PX_SDP_VIDEO_PROG_1_REF_ID,  
493|                                     PX_SDP_SERVICE_PROTECTION_ISMA  
494|                                     ));  
495|                                     var AccessTypeType v_AccessType :=  
496|                                     valueof(m_def_AccessType_bc_sdp(  
497|                                     f_getAccessType  
498|                                     (PX_BROADCAST_NETWORK_BEARER),  
499|                                     PX_SDP_VIDEO_PROG_1_REF_ID  
500|                                     ));  
501|                                     var PriceInfoType v_PriceInfo := valueof  
502|                                     (m_PriceInfo(  
503|                                     1,  
504|                                     PX_MONETARY_PRICE,  
505|                                     PX_CURRENCY  
506|                                     ));  
507|                                     var ServiceType v_Service := valueof  
508|                                     (m_def_Service(  
509|                                     PX_SGDU_SERVICE_ID,  
510|                                     v_FragmentVersion,  
511|                                     "PayTvChannel",  
512|                                     c_basicTv_ServiceType  
513|                                     ));  
514|                                     var ContentType v_Content := valueof  
515|                                     (m_def_Content(  
516|                                     PX_SGDU_CONTENT_ID_PROG_1,  
517|                                     v_FragmentVersion,  
518|                                     "Programme",  
519|                                     PX_SGDU_SERVICE_ID,  
520|                                     v_DateTime_StartTime,  
521|                                     v_DateTime_EndTime  
522|                                     ));  
523|                                     var ScheduleType v_Schedule := valueof  
524|                                     (m_def_Schedule(  
525|                                     PX_SGDU_SCHEDULE_ID_PROG_1,  
526|                                     v_FragmentVersion,  
527|                                     PX_SGDU_SERVICE_ID,  
PX_SGDU_CONTENT_ID_PROG_1,  
v_NTP_StartTime,
```

```

528         v_NTP_EndTime
529     ));
530
531     // TODO: Omitted here is setting of KMS
value - see open issues
532     var AccessType v_Access := valueof
(m_def_Access(
533         PX_SGDU_ACCESS_ID_PROG_1,
534         v_FragmentVersion,
535         v_AccessType,
536         PX_SGDU_SERVICE_ID,
537         PX_SGDU_SCHEDULE_ID_PROG_1,
538         c_class_SG
539     ));
540
541     var PurchaseItemType v_PurchaseItem :=
valueof(m_def_PurchaseItem(
542         PX_SGDU_PURCHASE_ITEM_ID,
543         v_FragmentVersion,
544         PX_SGDU_PURCHASE_DATA_ID,
545         PX_SGDU_SERVICE_ID,
546         "PurchaseItem1"
547     ));
548     v_PurchaseItem.Descriptions := {{lang :
= omit, text_:= "ISMACrypt"}}};
549
550     var PurchaseChannelType
v_PurchaseChannel := valueof(m_def_PurchaseChannel(
551         PX_SGDU_PURCHASE_CHANNEL_ID,
552         v_FragmentVersion,
553         "PurchaseChannel",
554         PX_PURCHASE_URL
555     ));
556
557     var PurchaseDataType v_PurchaseData :=
valueof(m_def_PurchaseData(
558         PX_SGDU_PURCHASE_DATA_ID,
559         v_FragmentVersion,
560         {{omit, "Discount"}}},
561         PX_SGDU_PURCHASE_ITEM_ID,
562         PX_SGDU_PURCHASE_CHANNEL_ID,
563         v_PriceInfo
564     ));
565
566     f_addServiceFragment(v_SGDU, v_Service);
567     f_addContentFragment(v_SGDU, v_Content);
568     f_addScheduleFragment(v_SGDU,
v_Schedule);
569     f_addAccessFragment(v_SGDU, v_Access);
570     f_addPurchaseItemFragment(v_SGDU,
v_PurchaseItem);
571     f_addPurchaseChannelFragment(v_SGDU,
v_PurchaseChannel);
572     f_addPurchaseDataFragment(v_SGDU,
v_PurchaseData);
573     f_addSdpFragment(v_SGDU,
v_FragmentVersion, v_Sdp);
574
575     f_main_ctrl_broadcastServiceGuide
(v_SGDU, omit);

```

```

528         v_NTP_EndTime
529     ));
530
531     // TODO: Omitted here is setting of KMS
value - see open issues
532     var AccessType v_Access := valueof
(m_def_Access(
533         PX_SGDU_ACCESS_ID_PROG_1,
534         v_FragmentVersion,
535         v_AccessType,
536         PX_SGDU_SERVICE_ID,
537         PX_SGDU_SCHEDULE_ID_PROG_1,
538         c_class_SG
539     ));
540
541     var PurchaseItemType v_PurchaseItem :=
valueof(m_def_PurchaseItem(
542         PX_SGDU_PURCHASE_ITEM_ID,
543         v_FragmentVersion,
544         PX_SGDU_PURCHASE_DATA_ID,
545         PX_SGDU_SERVICE_ID,
546         "PurchaseItem1"
547     ));
548     v_PurchaseItem.Descriptions := {{lang :
= omit, text_:= "ISMACrypt"}}};
549
550     var PurchaseChannelType
v_PurchaseChannel := valueof(m_def_PurchaseChannel(
551         PX_SGDU_PURCHASE_CHANNEL_ID,
552         v_FragmentVersion,
553         "PurchaseChannel",
554         PX_PURCHASE_URL
555     ));
556
557     var PurchaseDataType v_PurchaseData :=
valueof(m_def_PurchaseData(
558         PX_SGDU_PURCHASE_DATA_ID,
559         v_FragmentVersion,
560         {{omit, "Discount"}}},
561         PX_SGDU_PURCHASE_ITEM_ID,
562         PX_SGDU_PURCHASE_CHANNEL_ID,
563         v_PriceInfo
564     ));
565
566     f_addServiceFragment(v_SGDU, v_Service);
567     f_addContentFragment(v_SGDU, v_Content);
568     f_addScheduleFragment(v_SGDU,
v_Schedule);
569     f_addAccessFragment(v_SGDU, v_Access);
570     f_addPurchaseItemFragment(v_SGDU,
v_PurchaseItem);
571     f_addPurchaseChannelFragment(v_SGDU,
v_PurchaseChannel);
572     f_addPurchaseDataFragment(v_SGDU,
v_PurchaseData);
573     f_addSdpFragment(v_SGDU,
v_FragmentVersion, v_Sdp);
574
575     f_main_ctrl_broadcastServiceGuide
(v_SGDU, omit);

```


576			576		
577	f_main_ut_RunBCastApplication();		577	f_main_ut_RunBCastApplication();	
578	if (PX_GET_SG_DISPLAY_MANUALLY) {		578	if (PX_GET_SG_DISPLAY_MANUALLY) {	
579	f_main_ut_GetServiceGuide		579	f_main_ut_GetServiceGuide	
	(PX_SGDD_ID);			(PX_SGDD_ID);	
580	}		580	}	
581	f_main_ut_CheckService("PayTvChannel");		581	f_main_ut_CheckService("PayTvChannel");	
582	f_main_ut_PurchaseService		582	f_main_ut_PurchaseService	
	("PayTvChannel");			("PayTvChannel");	
583	f_main_ctrl_awaitingSubscription		583	f_main_ctrl_awaitingSubscription	
	("PayTvChannel"); // TODO: This function currently does not include any			("PayTvChannel"); // TODO: This function currently does not include any	
	checking and handling of authentication - see open issues			checking and handling of authentication - see open issues	
584	// TODO: Omitted here is generation of		584	// TODO: Omitted here is generation of	
	proper acceptance of terminal purchase request - see open issues			proper acceptance of terminal purchase request - see open issues	
585			585		
586	f_main_ut_SelectService("PayTvChannel");		586	f_main_ut_SelectService("PayTvChannel");	
587	f_main_ut_UseService("PayTvChannel");		587	f_main_ut_UseService("PayTvChannel");	
588	f_main_ut_CheckVideo("Programme");		588	f_main_ut_CheckVideo("Programme");	
589			589		
590	// postamble		590	// postamble	
591	// TODO: Omitted here is stopping the		591	// TODO: Omitted here is stopping the	
	streaming server to broadcast the encrypted TV Channel program since			streaming server to broadcast the encrypted TV Channel program since	
	handling of SEK delivery is open issue			handling of SEK delivery is open issue	
592	f_main_ut_PowerOff();		592	f_main_ut_PowerOff();	
593			593		
594	f_cf_Broadcast_down();		594	f_cf_Broadcast_down();	
595	f_cf_Interaction_down();		595	f_cf_Interaction_down();	
596	f_cf_UpperTester_down();		596	f_cf_UpperTester_down();	
597	} // end test case TC_BCAST_CONTPROT_CONF_103	<>	597	} // end test case TC_BCAST_INTER_CONF_103	
598	} // end group clientConformanceTestCases	=	598	} // end group clientConformanceTestCases	
599	} // end group bcastConformanceTestCases		599	} // end group bcastConformanceTestCases	
600	}		600	}	