

# **C6-210350 - Discussion on Seamless testing for NSR SIM devices**

**3GPP TSG-CT6 Meeting #109-e  
Online, 16th Nov 2021 - 19th Nov 2021**

**Apple , Comprion**

# Challenges

- In CT6 - WI (C6-210062) was created to develop a new test specification to include new test methods required for devices with Non removable -SIMs (eg: iSIM/eSIM, SSP/iSSP).
- With the usage of an eUICC only device , ETSI TS 102 671 (M2M UICC) form factor (MFF1/ MFF2) or ETSI TS 103 666-2 (iSSP) access to a physical connection in between UICC and terminal is not available. This implies all 31.124 / 31.121 tests that require content and/or availability checks on the APDU based 3GPP application (USIM) as part of the test procedure related to a conformance requirement cannot be tested and such devices will FAIL UE conformance Testing.
- Verification of APDU content has been a fundamental requirement for USIM / USAT testing.
- This challenge - needs to be addressed so that reliable Standard testing procedure can be adopted , guaranteeing seamless testing, device Security.



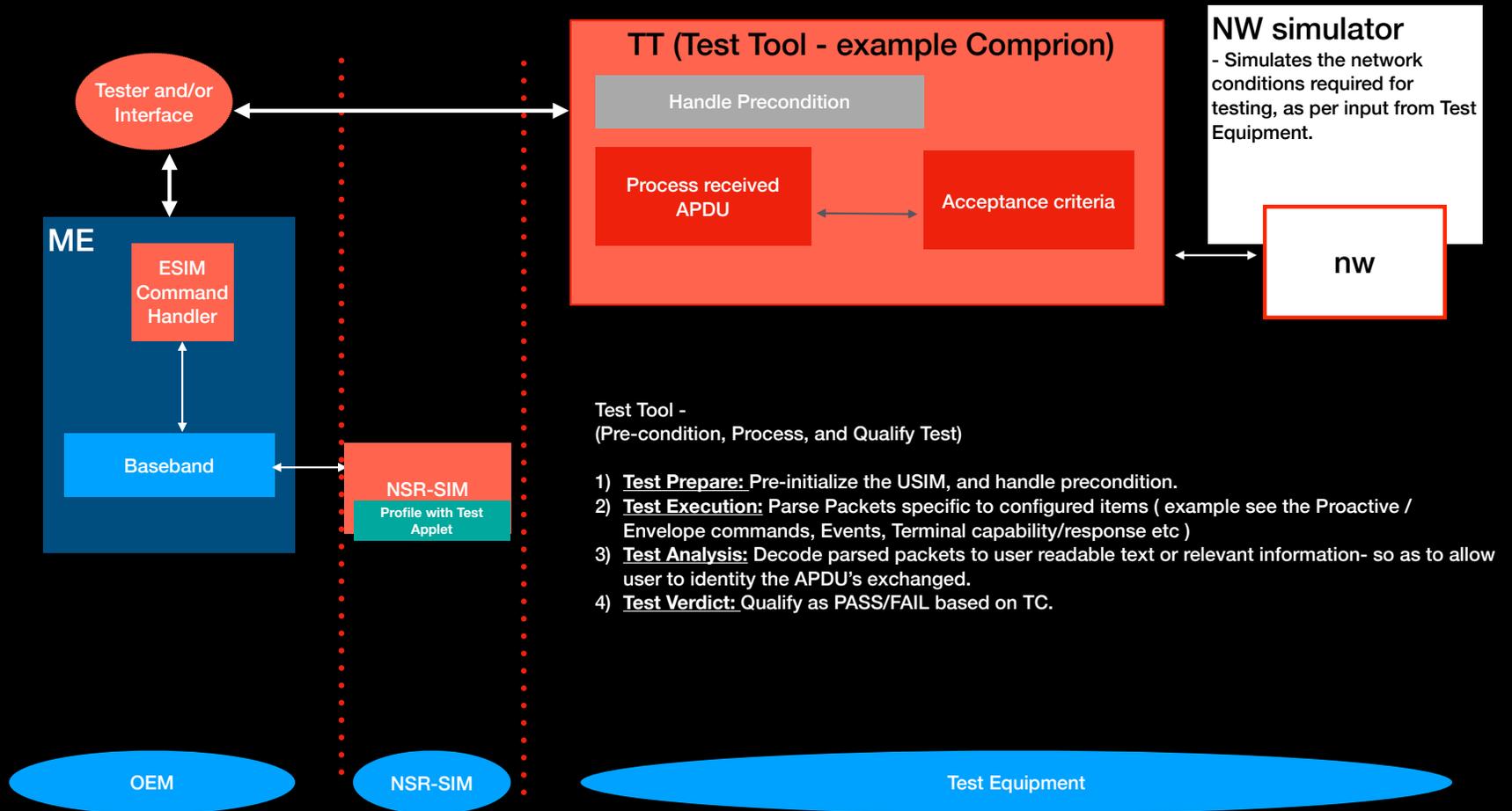
For example, this is how it can be done today on a removable sim devices

**Proposal**

# Seamless approach

- Device can give an interface to (i) the APDU traffic for seamless & explicit verification (ii) manage the eUICC per test needs.
- STK test profile , with predefined test DF / EF's for STK testing (covered in slide 8,9).

# Detailed Call Flow



# Requirements

## Test Equipment

- Develop Test case specific profiles.
- Pre-initialize the UE / Applet via RFM or user interaction.
- In case of manual user interaction , shall provide the tester the data to modify on eUICC/iSIM/SSP per test case.
- The commands shall follow the generalized structure defined in ETSI TS 102 221.
- Receive the raw command input from the DUT (manual/automated)
- Validate the received data/APDU traffic.

## eSIM Command Handler

- **Logging:** Direct Access to APDU traffic (between Baseband and eUICC/iSIM/SSP)
- **Interface:** Support the interface required to interact with external Test equipment.
- **Security:** This mechanism to monitor APDU traffic is only available for testing purpose, and disabled in production device.

## Applet / Profile

- May Implement the applet according to these defined Guidelines.
- Modified TS.48 profile for conformance testing.

# Advantages

Areas	Proposal
Security Impact	- Nil (Production card OS remains the same as test OS without any modifications)
Verification methods	- Applet - Explicit Verification possible. (Even without 4FF sim interface access). - Ability to verify the correctness of the APDU by the TT and UE.
OS impact	- No special OS required.
Test Case impact	- Medium
Adaptation	- Easy - Not limited to specific key, specific IMSI or Test Authentication Algorithm - No security impact, and no new Events required for verification.
Memory	- Nil

# Solution 1: Generic STK Applet

The Generic STK Test Profile allows a tester to configure the proprietary test EF's with specific data.

The Applet monitors these EF's and triggers appropriate STK events and Proactive commands.

## Basic Overview

A special DF called (AppletTest) is defined, with few Elementary files (EF) under this DF as shown. The purpose of these Elementary files is to hold the data configured by the tester.

```
MF
|
|--DF _AppletTest (7FXX)
|   |
|   |--EF_STK (XX XX)           // Sets Proactive command in records
|   |--EF_setSTK (XX XX)       // Triggers the set Proactive command from specific record
|   |--EF_CC (XX XX)           // Call control response
|   |--EF_EventList (XX XX).    // Specifies the configured events in sequence
```

## Tester's Responsibility

The tester configures the required EF's for the required STK / Event APDU , so that it can be sent to the terminal as per the 3GPP/ETSI specifications. (3GPP 31.111 or ETSI TS 102 223)

## Applet

The applet utilizes the data configured in these new EF's so as to trigger the specified command from UICC.

**Note:** The STK test applet can ignore the provided capabilities of the terminal during initialization. The STK applet can ignore the contents of the Terminal response.

A special DF called (STKTest) can be defined, with few Elementary files under it. The purpose of these Elementary files is to hold the data configured by the tester.

For example :-

(1) EF\_STK - is a linear fixed file with 255 bytes record, and multiple records (for example , say 40. Can use more if needed).

: Each record can hold the tester Configured APDU data, specific to the Proactive command that needs to be issued to the Terminal by the tester (as per 3GPP 31.111 or ETSI TS 102 223).

(2) EF\_setSTK - This EF\_setSTK is one byte long and can hold the record number of EF\_STK that needs to be triggered.

If EF\_setSTK is configured with '0x01', The applet will only issue the command (APDU) configured at specified record 01 of EF\_STK. Once the command is issued successfully, the applet will clear this EF\_setSTK. For example , If EF\_STK is configured for Open Channel at record 1, and EF\_setSTK is configured as 00000001 - then the applet will only trigger Open channel command.

If EF\_SetSTK is configured as (0x80) ( a specific configuration ), then this will imply to trigger all the STK commands from all configured records in EF\_STK. For example, if EF\_STK is configured with multiple STK commands at individual records 1 to 10, and EF\_setSTK is configured as 10000000 - then the applet will trigger the STK commands sequentially from 1 till 10 from EF\_STK records 1 to 10. This is helpful when we want to issue Proactive Commands in configured sequence.

(3) EF\_CC - is transparent file with length 255 bytes.

This file holds the Call control Response as configured by the tester to process the call control commands based on the CC configured in UST.

(4) EF\_EventList - It is a transparent file that is able to hold 255 bytes, unused bytes can be set to FF.

Its of format <length><event1><event2>...FF FF., where length is the number of configured events. Supported events are defined in TS102.223 and 3GPP 31.111 Sections 8.25. Based on the event list(s) configured in this EF, the Applet will issue that specific Event(s) in sequence at the beginning of the tool kit session. This configured data persists even after card reset, until cleared by the tester. (The tester can clear these events by manually updating this file contents with all FF's).

# Example - TC 27.x Open/Close Channel TC

Legacy Test Spec - Current for Removable SIM

Step	Direction	MESSAGE / Action	Comments
1	USER → ME	Set and configure URSP rules with DNN "TestGp.rs" in the terminal configuration if required. Always-on PDU session using DNN "internet" is configured in the terminal.	[see initial conditions]
2	ME → NG-SS	ME successfully REGISTER with NG-RAN cell.	
3	ME → NG-SS	Always-on Internet PDU Session is established successfully.	
4	UICC → ME	PROACTIVE COMMAND PENDING: OPEN CHANNEL 4.2.1	
5	ME → UICC	FETCH	
6	UICC → ME	PROACTIVE COMMAND: OPEN CHANNEL 4.2.1	
7	ME → USER	The ME may display channel opening information.	
8	ME → NG-SS	PDU SESSION ESTABLISH REQUEST is sent to the network.	DNN=TestGp.rs, S-NSSAI='01 01 01 02', SSC mode=2.
9	NG-SS → ME	PDU SESSION ESTABLISH ACCEPT	
10	ME → UICC	TERMINAL RESPONSE: OPEN CHANNEL 4.2.1	[Command performed successfully]
11	UICC → ME	PROACTIVE COMMAND PENDING: CLOSE CHANNEL 4.2.1	
12	ME → UICC	FETCH	
13	UICC → ME	PROACTIVE COMMAND: CLOSE CHANNEL 4.2.1	
14	ME → UICC	TERMINAL RESPONSE CLOSE CHANNEL 4.2.1	[Command performed successfully]
15	USER → ME	Wait 30 seconds, then switch off the terminal	

Proposal - Simple modification to handle and verify e-ONLY device

Step	Direction	MESSAGE / Action	Comments
1	USER → ME	Set and configure URSP rules with DNN "TestGp.rs" in the terminal configuration if required. Always-on PDU session using DNN "internet" is configured in the terminal. Configure eUICC with the Open Channel Proactive command in EF_STK at record 1, and Close Channel Proactive command at EF_STK record 2.	[see initial conditions]
2	ME → NG-SS	ME successfully REGISTER with NG-RAN cell.	
3	ME → NG-SS	Always-on Internet PDU Session is established successfully. Configure the EF_set_STK to trigger record 01.	
4	eUICC → ME	PROACTIVE COMMAND PENDING: OPEN CHANNEL 4.2.1	
5	ME → eUICC	FETCH	
6	eUICC → ME	PROACTIVE COMMAND: OPEN CHANNEL 4.2.1	
7	ME → USER	The ME may display channel opening information.	
8	ME → NG-SS	PDU SESSION ESTABLISH REQUEST is sent to the network.	DNN=TestGp.rs, S-NSSAI='01 01 01 02', SSC mode=2.
9	NG-SS → ME	PDU SESSION ESTABLISH ACCEPT	
10	ME → eUICC	TERMINAL RESPONSE: OPEN CHANNEL 4.2.1 Configure the EF_set_STK to trigger record 02.	[Command performed successfully]
11	UICC → ME	PROACTIVE COMMAND PENDING: CLOSE CHANNEL 4.2.1	
12	ME → eUICC	FETCH	
13	eUICC → ME	PROACTIVE COMMAND: CLOSE CHANNEL 4.2.1	
14	ME → eUICC	TERMINAL RESPONSE CLOSE CHANNEL 4.2.1	[Command performed successfully]
15	USER → ME	Wait 30 seconds, then switch off the terminal	

# Addressing Qualcomm concerns

What is not acceptable?	Why?	Response
<p><b>1</b> Baseband logging for APDU verification</p>	<ul style="list-style-type: none"> <li>• Test verdict depends on BB logging data and not on the true content of APDU commands/responses in SIM-ME interface</li> <li>• No standardized logging interface is available.</li> <li>• Loss of log packets can lead to test failures even device meets the spec requirements.</li> <li>• Some OEMs and Carriers are mandating certain logging data to be masked due to security reasons.</li> </ul>	<p>No security risk is there, since this is similar to the legacy case (using removable UICC/Test Equipment) - except that the logging is now done on DUT side.</p> <p>Logging interface is to be handled by DUT, so implementation specific - and OEM shall ensure that required 'minimal' data is guaranteed without loss for testing purpose if this method of testing is supported.</p>
<p><b>2</b> Test verdicts depend on tools within the DUT</p>	<ul style="list-style-type: none"> <li>• Test verdict shall not be determined based on data provided from the tools (eg: Logging tools) within DUT.</li> <li>• Test system shall use its own applications (eg: Test Applets) to determine test verdicts.</li> <li>• Otherwise DUT vendor can update/manipulate their own tool to meet the test case requirements.</li> </ul>	<p>Existing PTCRB/GCF test's already utilize DUT , Modem based solutions (example MMI, AT command), and this has never been a concern for PTCRB/GCF.</p> <p>Monitoring C-APDU , or R-APDU rely on the direct interface to NSR UICC or relevant software component. Similar concern of manipulation shall apply to even test methodology currently proposed by Qualcomm., including Test Events based solution.</p>
<p><b>3</b> A test solution without supporting all NSR SIM variants</p>	<ul style="list-style-type: none"> <li>• Test solution shall include support for devices with iSIM and iSSP.</li> <li>• APDU verification at SIM end is an important part for iSIM/iSSP solutions since multiple SW layers and processors are involved in the interface between device and iSIM/iSSP.</li> <li>• Baseband logging based solution does not verify complete APDU communication between device and iSIM/iSSP.</li> </ul>	<p>This method can be adopted for other NSR type devices as well, as the monitoring the APDU traffic can be done at the required interface (relevant software component as per implementation).</p>
<p><b>4</b> Test specification changes to include Test EF data required for STK test cases</p>	<ul style="list-style-type: none"> <li>• Number of Test EFs required and the content/format of data in those EFs depends on the Test Applet implementation.</li> <li>• As long as Test Applet can send command or response data as specified in the Test specification it should be sufficient.</li> <li>• Test Applet and Test EFs shall be implementation specific.</li> </ul>	<p>Agree, this can be implementation specific. However, a standardized approach in the definition of test EF can be helpful to ensure a well defined test specification.</p>