TTCN3 Experiences for LTE MME Call processing testing

ETSI TTCN-3 User Conference 2010

Motorola India Pvt Ltd., Bangalore, India

Prashant Ramaswamy (prashant.r@motorola.com)

Alwar Ethiraj (a15295@motorola.com)

AGENDA

- Product Overview
- Why TTCN3
- Motorola LTE TTCN3 Test Strategy
- Design Objectives
- Challenges
- Take-away from the experience

The Product - Introduction

LTE (Long Term Evolution) defines the 4th generation (4G) design for radio networks, introducing over the air speeds of the order of 100 Mbps.

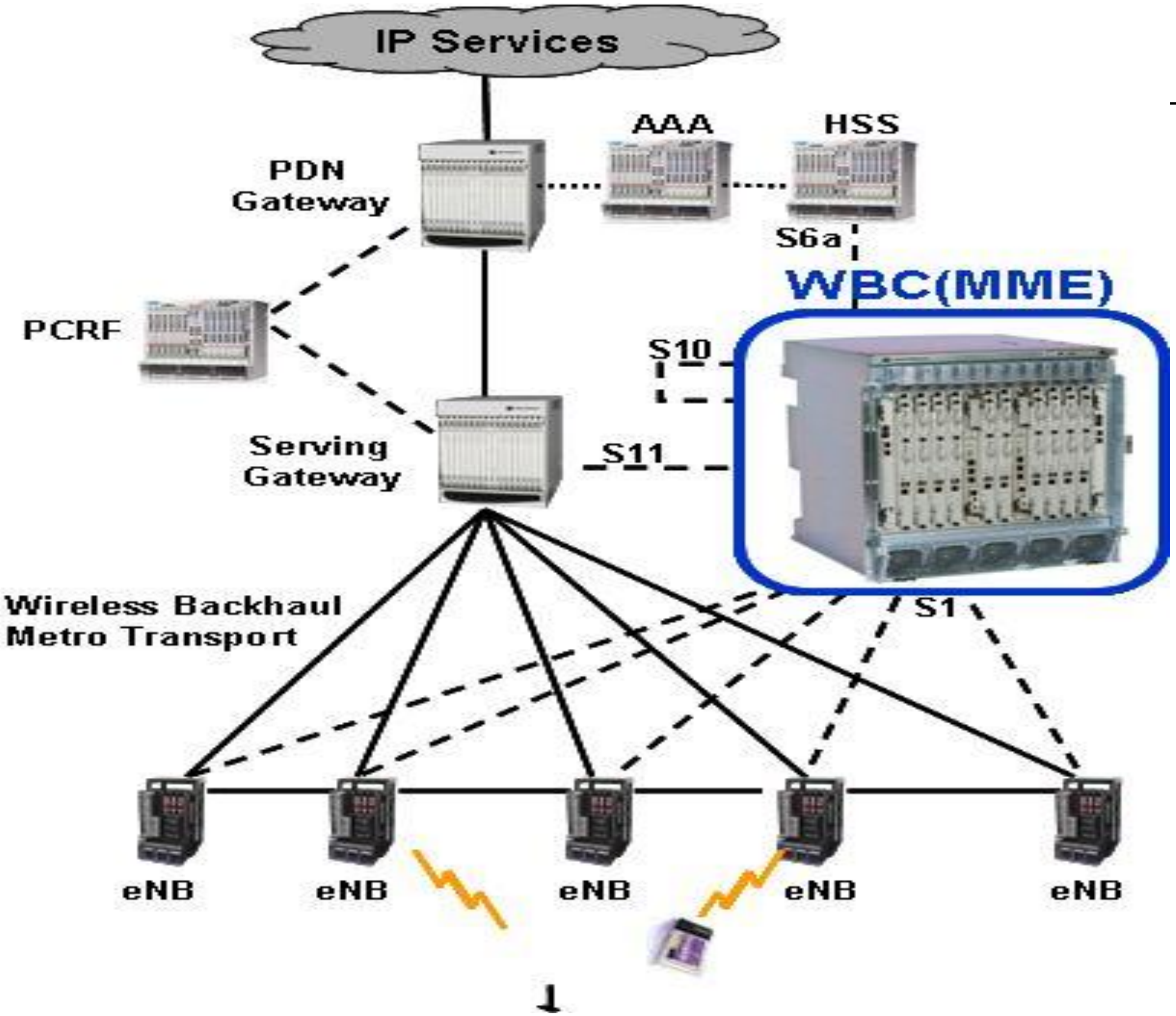
Motorola is one of the pioneers in developing commercially deployable end-to-end LTE solution.

LTE – Peak Rates

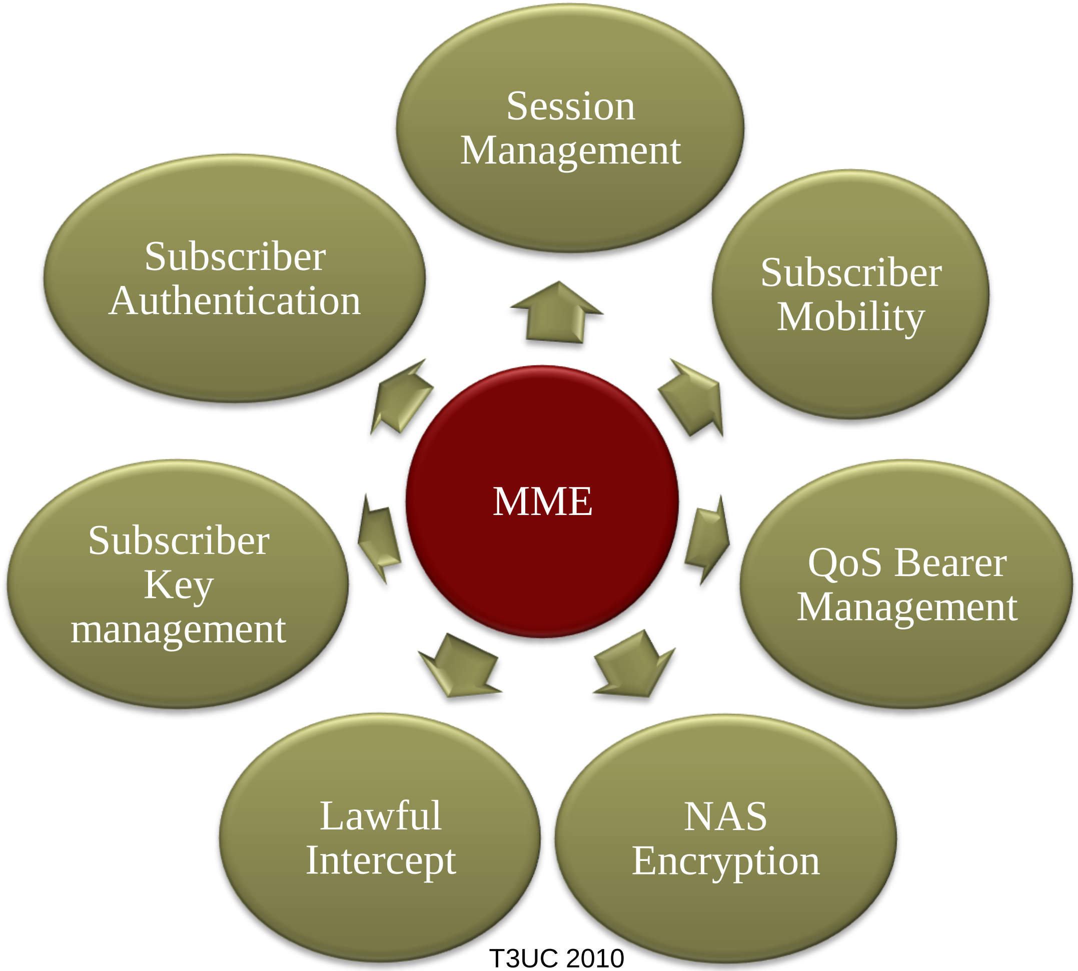
Antenna Technology		Bandwidth		
TYPE	QTY	5 MHZ	10 MHZ	20 MHZ
MIMO	2x2	29	59	117
MIMO	4x4	55	113	226

LTE Peak Data Rates (Mbps) - 5/6 Error Rate Coding

System Under Test



LTE Network – Functionality of SUT



Why TTCN3

Call Processing (CP) team test requirements:

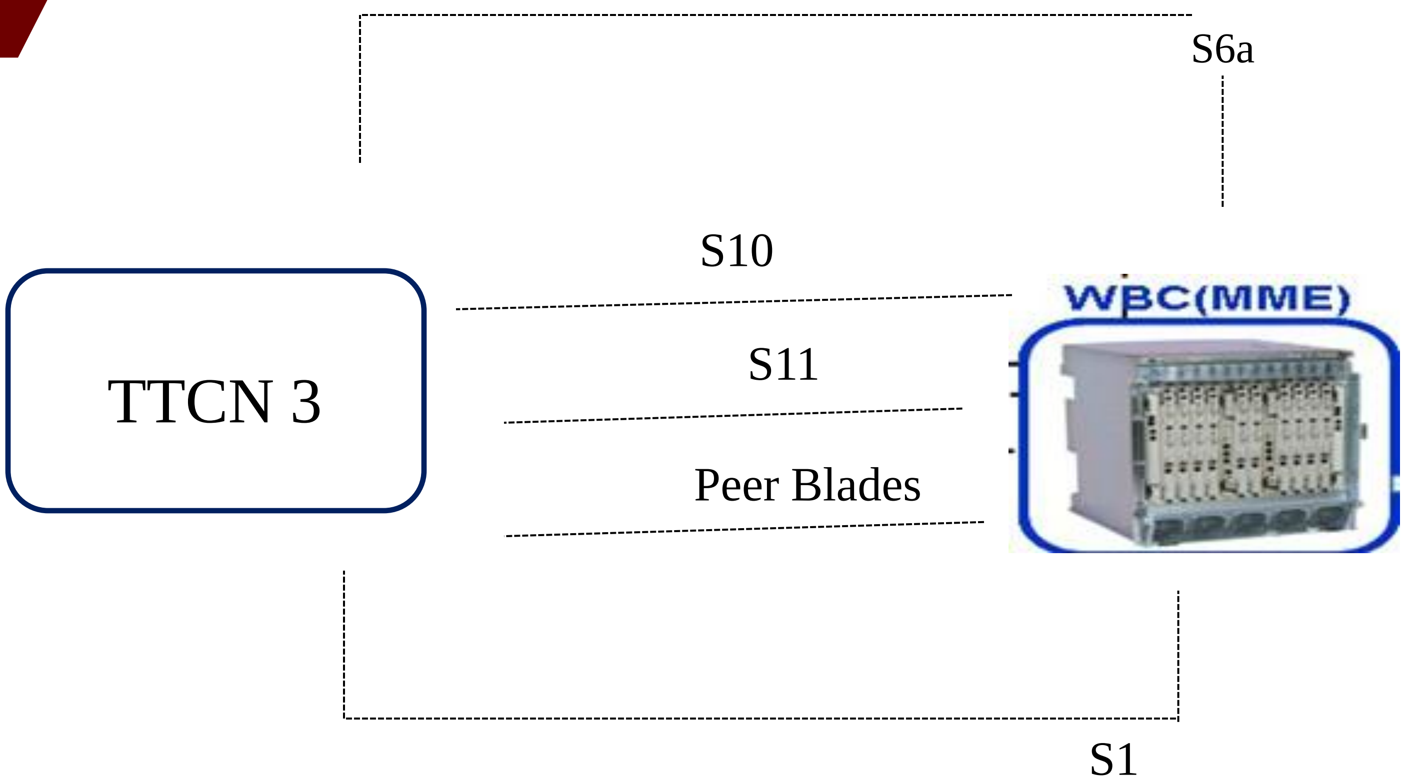
Testing on host (PC) and target	TTCN3 allows test Case reusability across phases
Automated testing and nightly regression for Agile development	High level of Automation & Regression Capabilities
Optimum test software development effort	Increased productivity due to availability of home grown re-usable components tailor-made for TTCN3 (Eg: Data type generator from iSL definitions)

Other TTCN3 benefits

- Standard language for test specification, with built-in capabilities of verdict handling
- Prior successful case studies

TTCN3 Solution

The Motorola MME solution runs on an ATCA 1440 Chassis with payload cards for Call processing, network termination and Shelf controller.

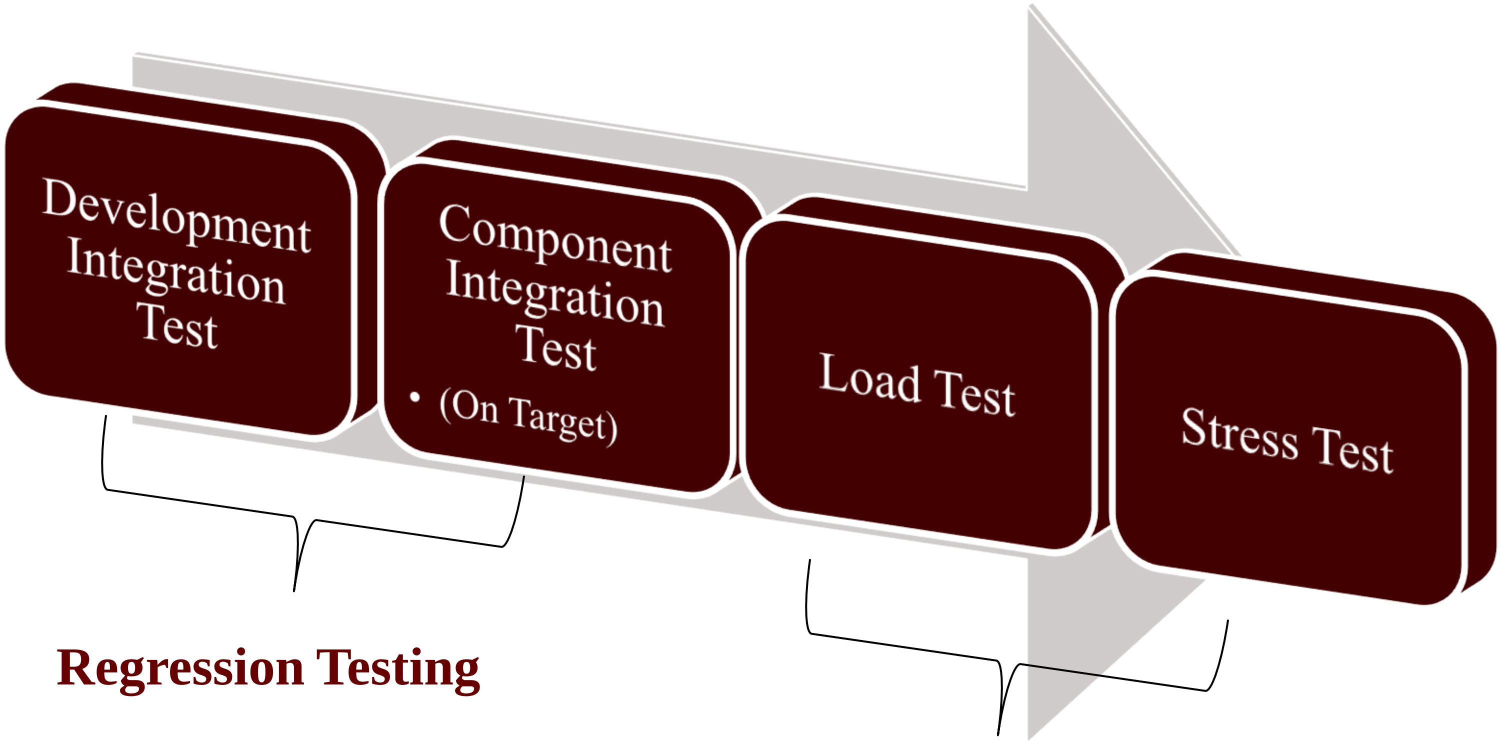


Motorola LTE TTCN-3 Test Strategy

- MME Call Processing (CP) functionality in Motorola is developed and tested using Model Driven Engineering.
  - TTCN-3 based CP Test system is developed to test MME behaviour and its various internal and external interfaces.
  - NAS, S1, S11, S10 and S6a are the external interfaces which are supported by the TTCN-3 test system.

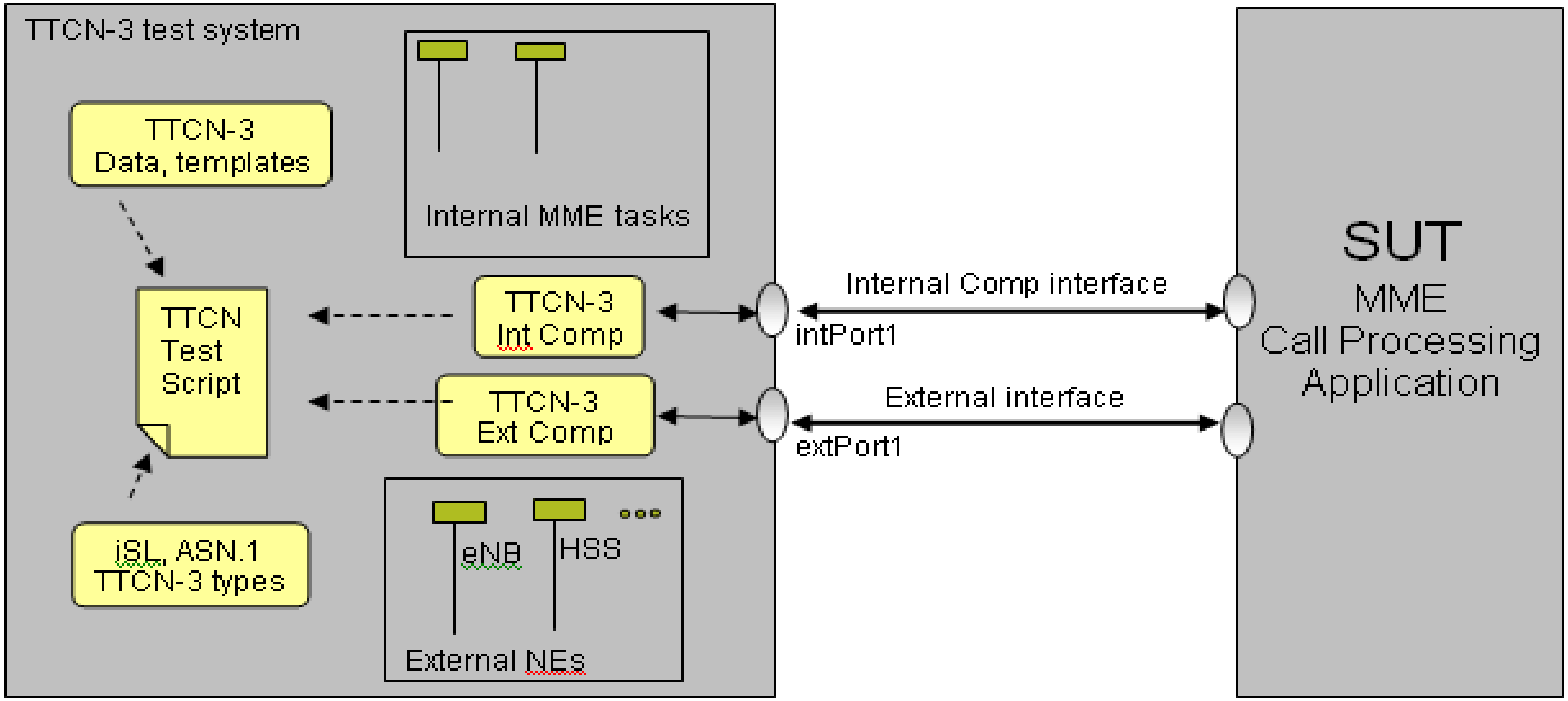
Motorola LTE TTCN-3 Test Strategy

Testing Scope

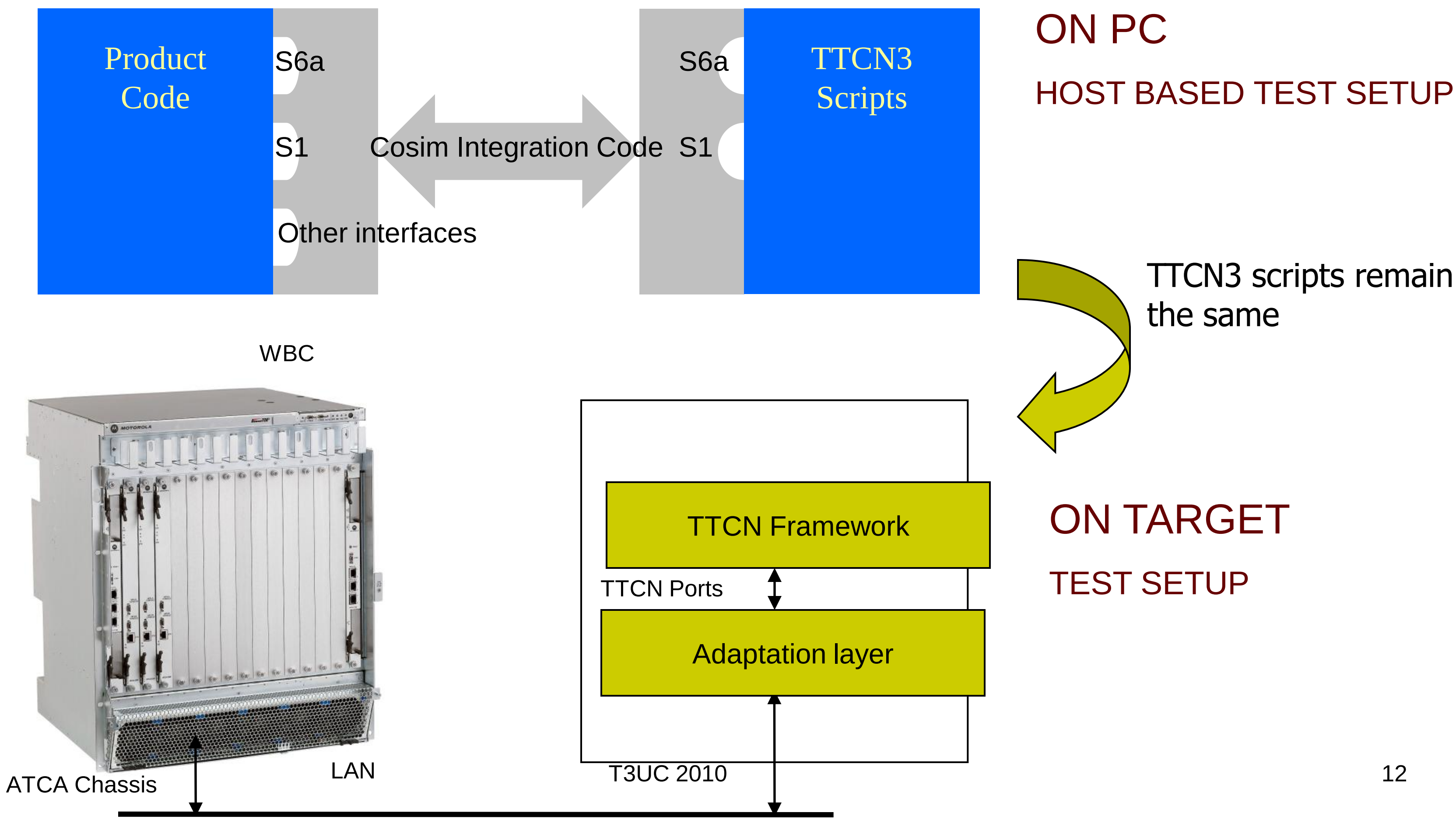


Motorola LTE TTCN-3 Test Strategy

GENERIC TEST SETUP



Motorola LTE TTCN-3 Test Strategy



Design Objectives & Challenges

- Design Objectives
  - Maintainable and Consistent Test System
  - Reusable scripts across phases
  - Ease of script development
- Challenges
  - Using third party codecs
  - NAS encoding
  - Constant 3GPP spec updates

Design Objectives

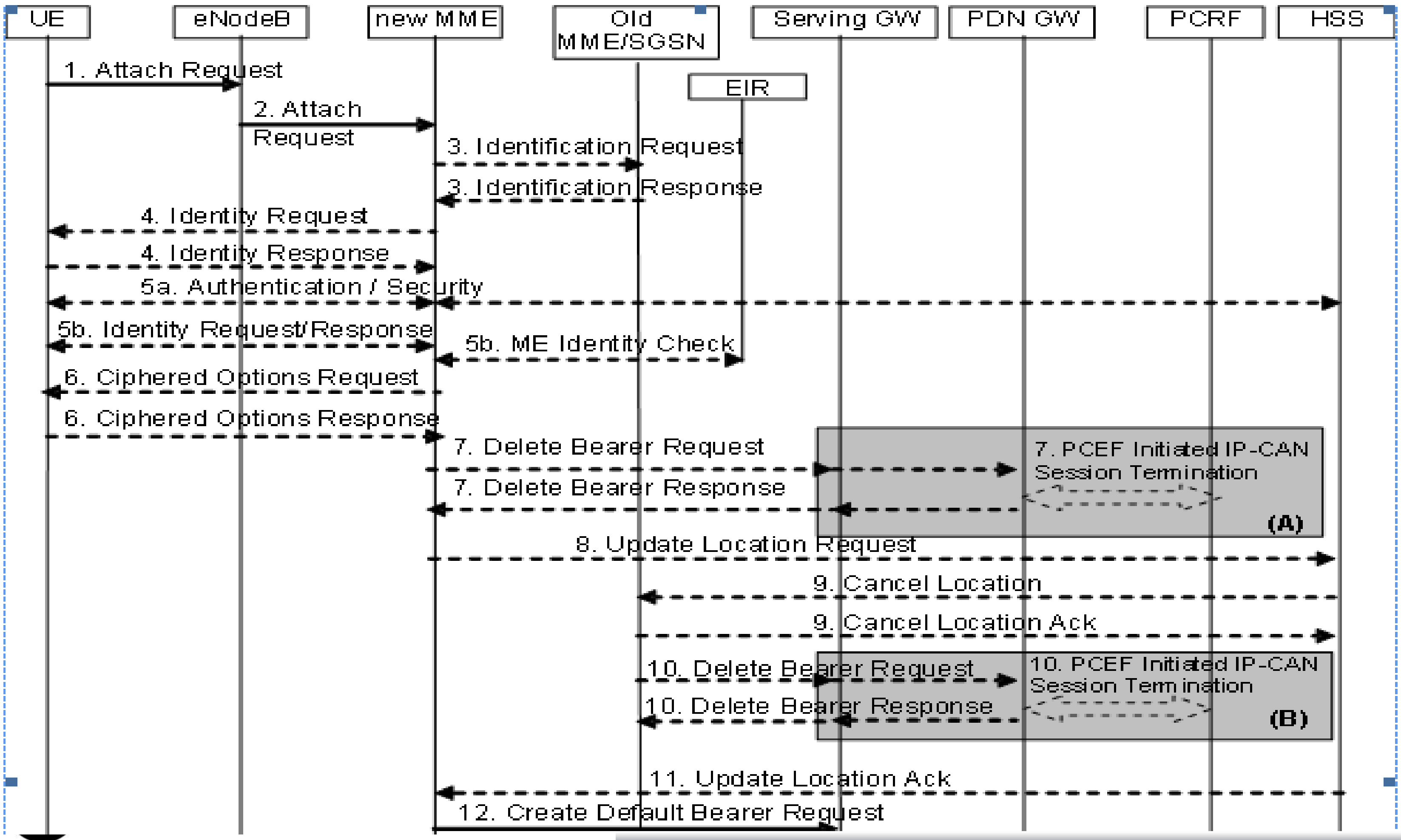
- Maintainable and Consistent Test System

Simple v/s Concurrent TTCN3

- WAB Call Processing involves many interfaces. Due to the non-deterministic behavior, call flow will involve different possible sequences.
  - In Simple TTCN, this kind of alternatives will result in complex non-concurrent test case descriptions. Maintainability and Flexibility are issues.
  - Concurrent TTCN solves this issue through the Parallel Test Components.

Sample Call flow

Simple v/s Concurrent TTCN3
- Typical Signaling Call Flow (Sample)



Source: 3GPP specification: 23.401 Ver 8.4.1 (http://www.3gpp.org/ftp/Specs/html-info/23401.htm)

Design Objectives

Reusable scripts across phases

- Writing reusable scripts across phases
  - Test Phase specific logic (eg. Execution of stubbed code in UT against actual code in IT) added to test case while developing the test.
  - Test Phase passed as test argument

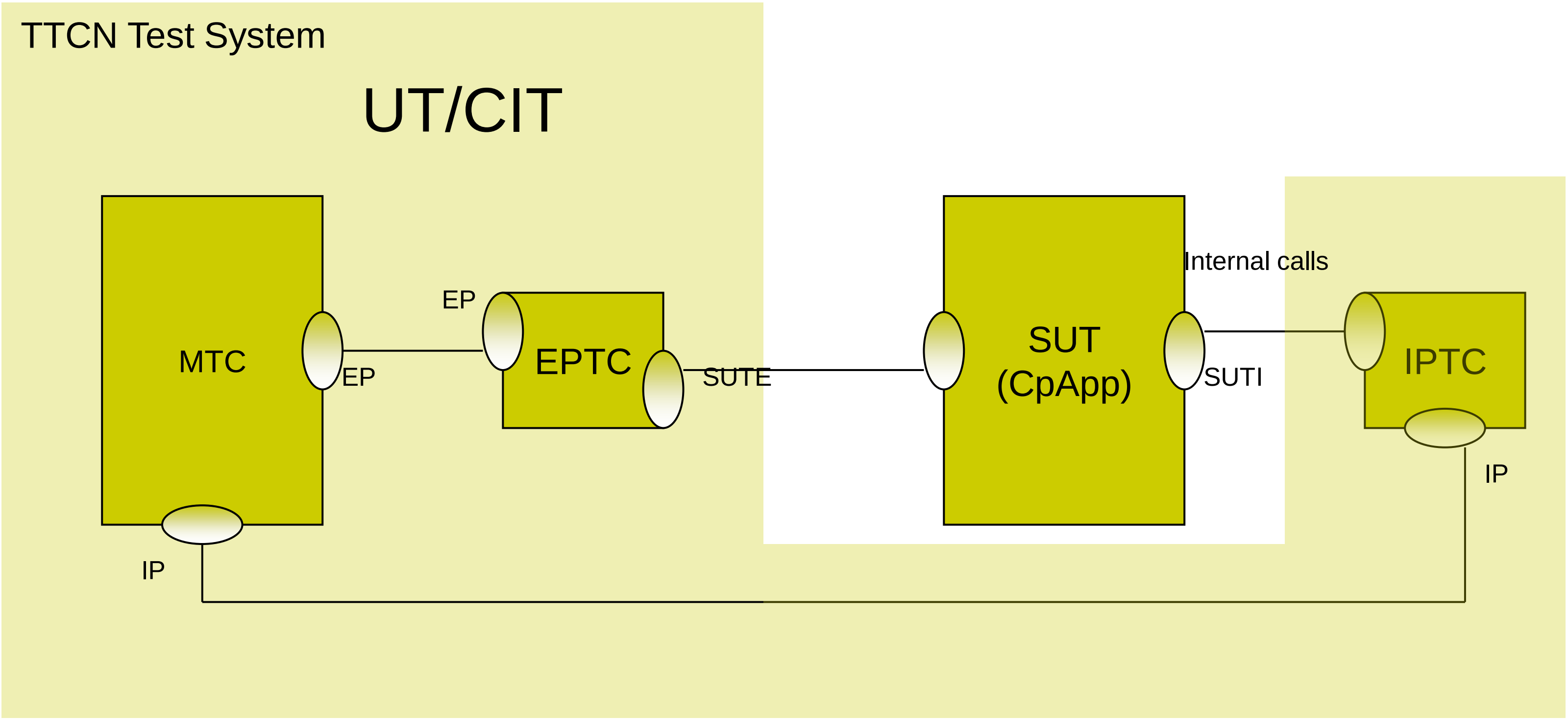
Design Objectives

Reusable scripts across phases – Test Sample

```
function ExtComp_Behaviour runs on ExternalComponent {
      alt {
         [] EP.getcall(send_attach:{phase}) {
            SUTE.send(attach_req);
         }
         repeat; // re-evaluate alt statement
      }
      [] TP.getcall(identity_exchange:{phase}) {
      …
      }
   }
   function IntComp_Behaviour runs on InternalComponent {
      alt {
         [] IP.getcall(update_location:{phase}) {
         if (phase == UT) {
            SUTI.receive(update_location);
            SUTI.send(update_location_ack);
         }
         repeat;
      }
   }
 testcase attach_test1(integer phase) runs on MTC system TSI {
      Initialization(mySys);
      EP.call(send_attach:{phase}, nowait);
      IP.call(update_location:{phase}, nowait);
      :
      endPTCs();
   }
```

Design Objectives

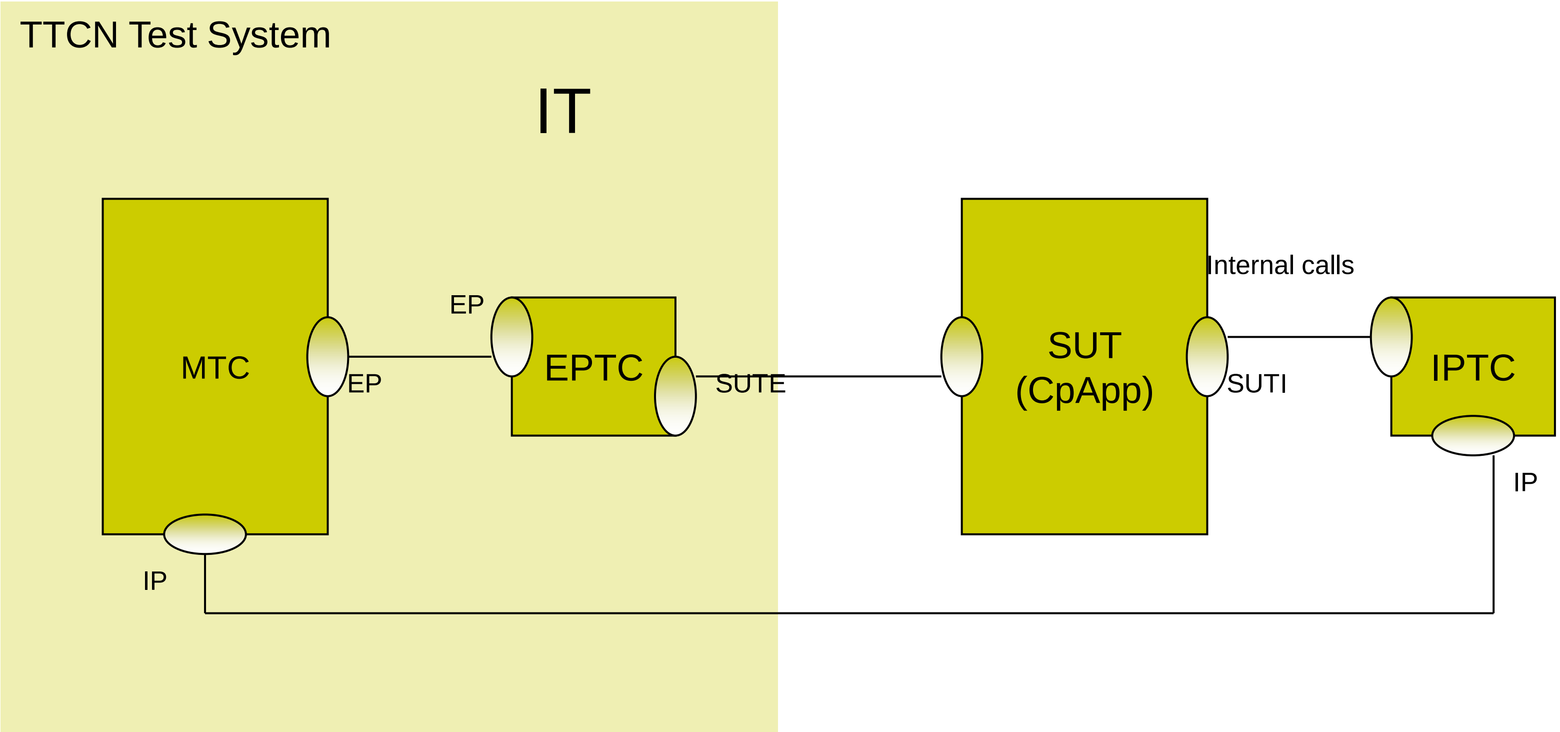
Reusable scripts across phases



EPTC => External Component
IPTC => Internal Component/Stubs

Design Objectives

Reusable scripts across phases



EPTC => External Component
IPTC => Internal Component/Stubs

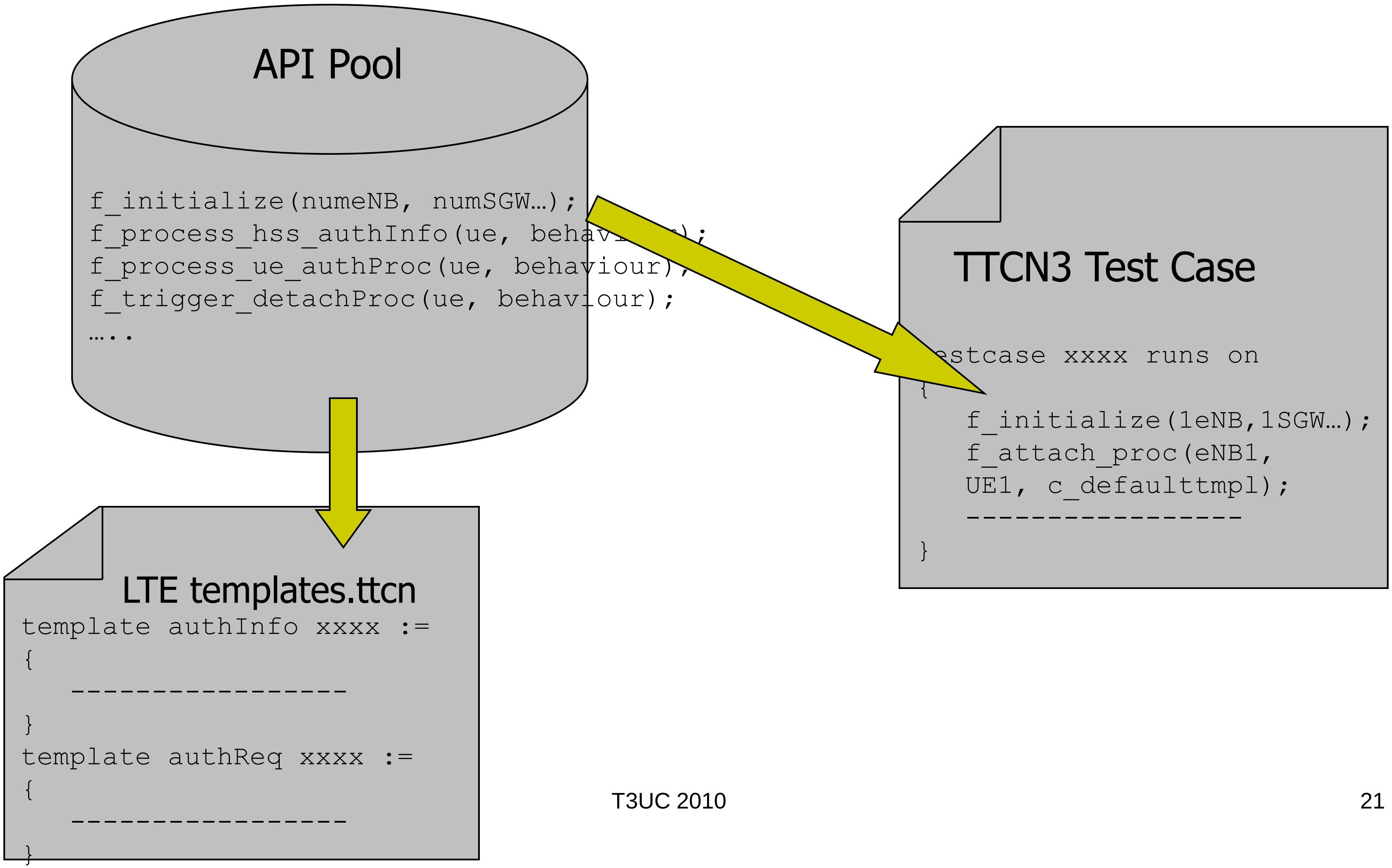
Design Objectives

Ease of script development and maintenance

- APIs, to develop script layer
  - Objective: To minimize templates in the test case.
  - APIs developed for script developers for every send and receive operation for an interface.
  - Option given to script developer to either use the APIs at the test case (suffices to handle 90% of scenarios) or add the actual signal send and receive statements.
    - The entire message can also be passed as argument to the API. The API (in the PTC) takes care of populating parameters, like sequence number etc, derived from previous input messages.

Design Objectives

Ease of script development and maintenance



Challenges
– Adaptation layer

- Third party codecs
  - For the S1 interface, the 3GPP specifications provide the ASN.1 definitions. Although TTCN-3 native language supports ASN.1 encoding, the TTCN tool did not support the right codec generation based on the 3GPP ASN.1 definitions. We opted to adopt a third party ASN.1 codec set
  - The S6a interface involves the Diameter stack over TCP/IP. We opted to plug-in a third party diameter stack
  - The adaptation layer was designed as below:
    - The external function written to check if there is a user defined encoder/decoder for each message and then to use the codec.
    - For every message, the IEs are populated and fed to call the third party auto-generated codec function (and vice versa; for decoding, the message structure is populated by passing the bitstream). This code was hand written.

Challenges
– NAS Encoding

- The LTE S1 interfaces involves encoding a PDU (NAS) as an information element inside an S1 PDU. This involves encoding the NAS PDU first, and then passing it as payload to the actual encoder (ASN.1 codec).
- TTCN3 facilitates this by providing the option for External C function calls. The NAS encoding is performed by making an external call to the C encoding function for the NAS message.

Challenges
– NAS Encoding

```
template NAS_IdentityRequest_t
     a_NAS_IdentityRequest_IMSI :=
{
    NASEmmHeader := a_NASEmmHeader_deflt,
    IdentityType :=
    {
        TypeOfId := IMSI
    }
};
```

External Encoder Function

Octetstring

```
Nas_msg := '0755428577356'O;
```

```
template DownlinkNASTransport
DownlinkNASTransport_tpl(ENB_UE_S1AP_ID
enbue_s1ap_id, template NAS_PDU nas_msg) :=
{
    mme_ue_s1ap_id := (0..2147483647),
    enb_ue_s1ap_id := enbue_s1ap_id,
    nas_pdu := nas_msg,
    handoverrestrictionlist :=
    {}
}
```

Challenges
- Constant 3GPP spec updates

- Constant 3GPP spec updates
  - Accommodating the constant churn and being compliant with a given version of specification using any other tool would have been an expensive effort. It would involve continuous time and effort in revisiting and updating the data types and templates and the test system might not be able to keep up with corresponding iteration of the new specification compliant SUT.
  - This challenge was absorbed in a smooth way using TTCN3, with the major enabler being home grown Data type generation tools. The tool generates TTCN data types, given the ASN.1 definitions published by 3GPP.
  - This prevented downtime of the test system at any point of time.

Our experience - Your take-away

Concurrent TTCN3

- **Uncluttered test cases**
  - Ideal option when the PTCs are relatively independent and do not need intervention from MTC.
  - When majority of the test control lies with MTC, more inter PTC communication is involved. This adds to the complexity when debugging test script issues.

API based Script development

- **Reduced time to market**
  - Provides flexibility for script developer to write test cases without delving into intricacies.
  - Update of TTCN3 templates owing to specification and definition update is limited to a single location (template file).

Test Case Reusability Across Phases

- **By incorporating the approach during design phase itself.**

Our experience - Your take-away

Maintenance of third party codecs

- **Opt for built-in Codecs or Plug-ins wherever possible.**
  - In case of need to integrate third party codecs, budget for integration effort.

NAS Encoding

- **TTCN3 External C Function calls**

Constant 3GPP Specifications update

- **TTCN3 Data type auto-generation tools give almost zero down-time of test system.**

THANK YOU