Automated Target Testing with TTCN-3: Experiences from WiMAX Call Processing Features

By
Bhaskar Rao G
Srinath Y
Sridhar Y
Jitesh M

Motorola India Pvt Ltd, Hyderabad
bhaskarraog@motorola.com

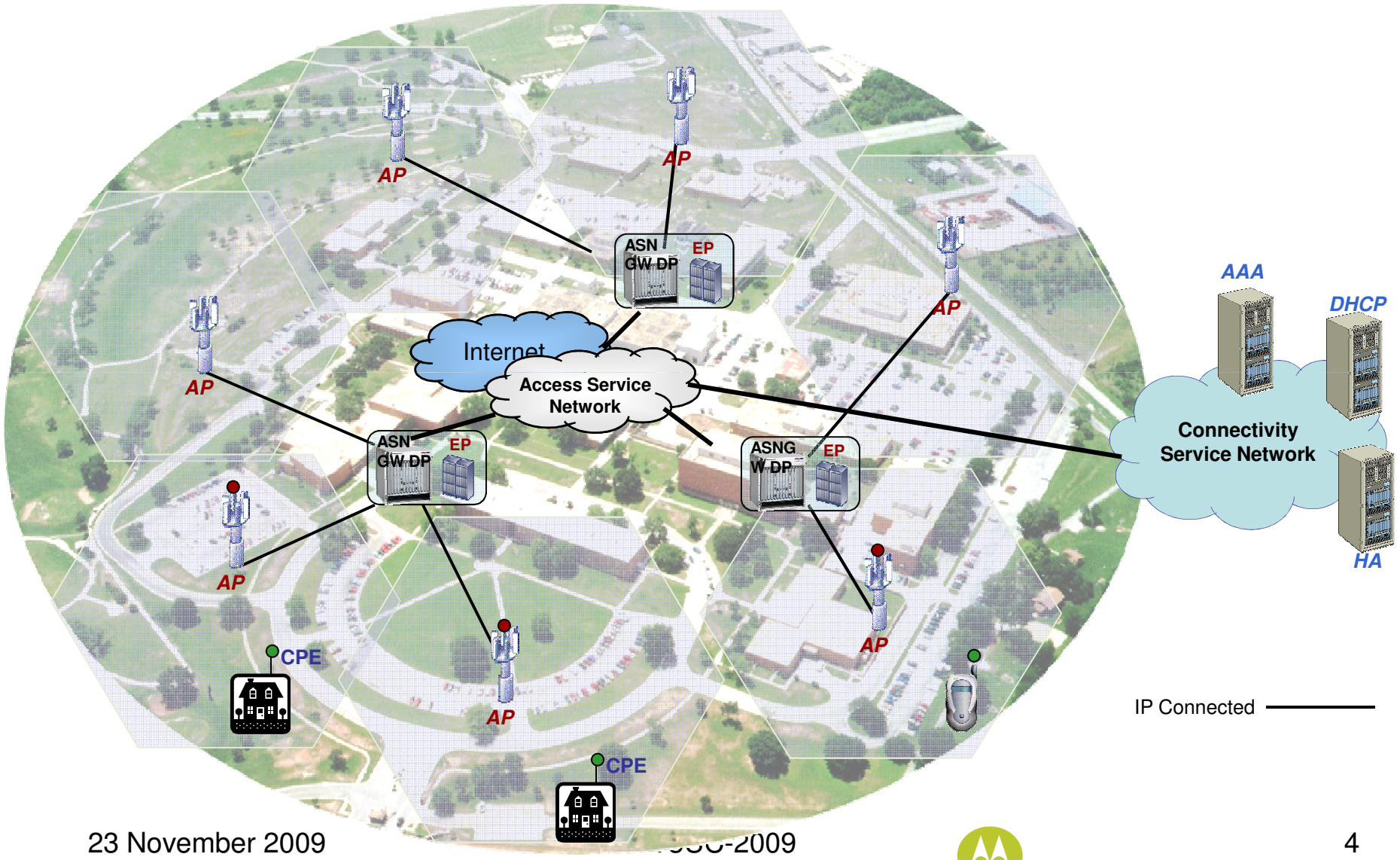
Agenda

- What is WiMAX
- WiMAX deployment and architecture
- ASNGW Decision Point feature architecture
- Test set-up for DP testing
- Motivation for TTCN-3
- Automation framework
- Feature Testing Strategy
- Test system details
- Test Execution set-up
- Challenges faced
- Benefits
- Take-away
- Conclusions

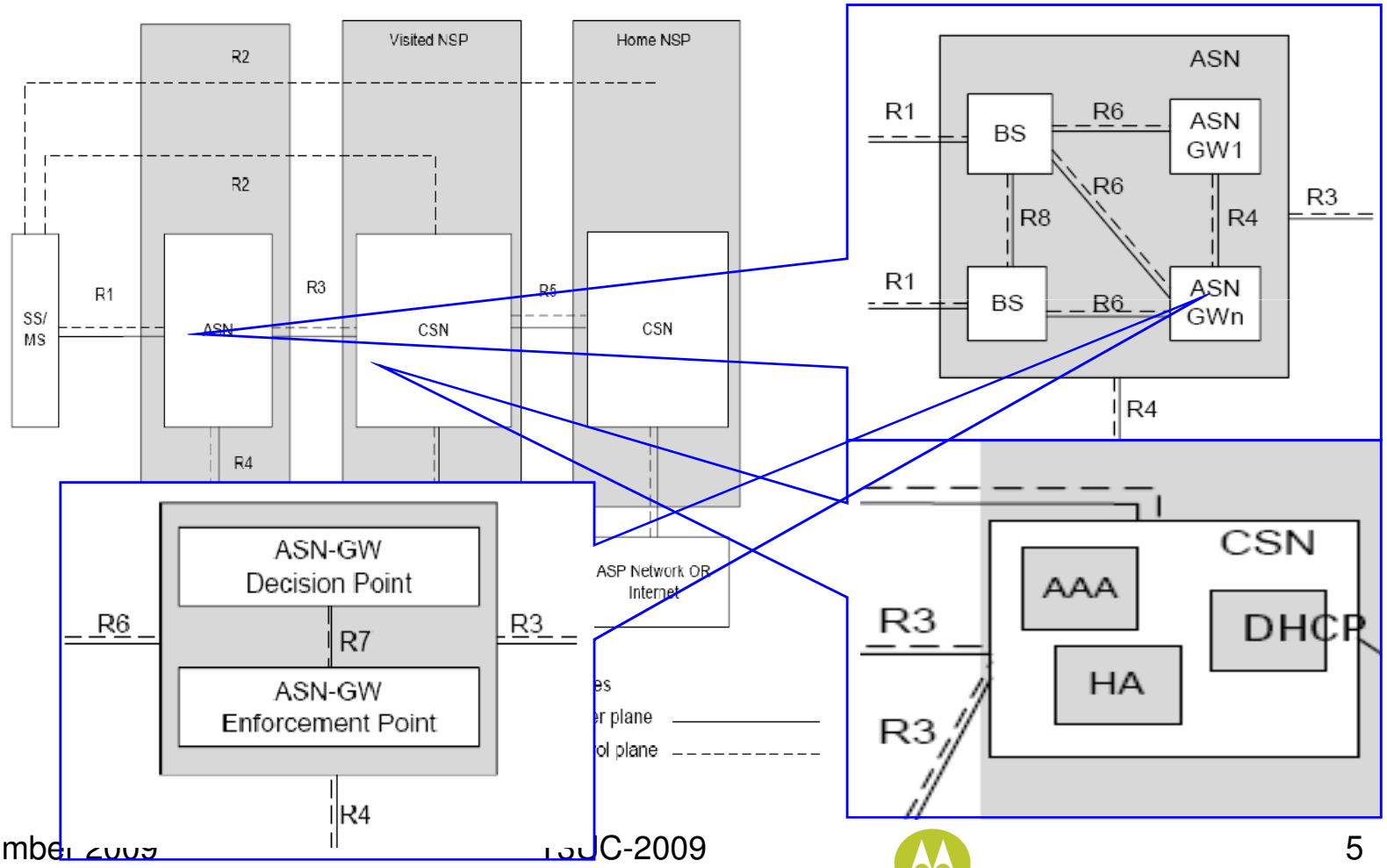
WiMAX

- Standards based technology - IEEE 802.16e
- Superior Performance – 75Mbps of throughput
- Scalable channel bandwidths - 12.5 to 20MHz
- Mobility of subscribers in the network coverage is supported for speeds ranging between 60kmph to 120 kmph
- Audio/Video streaming during mobility

WiMAX: Deployment



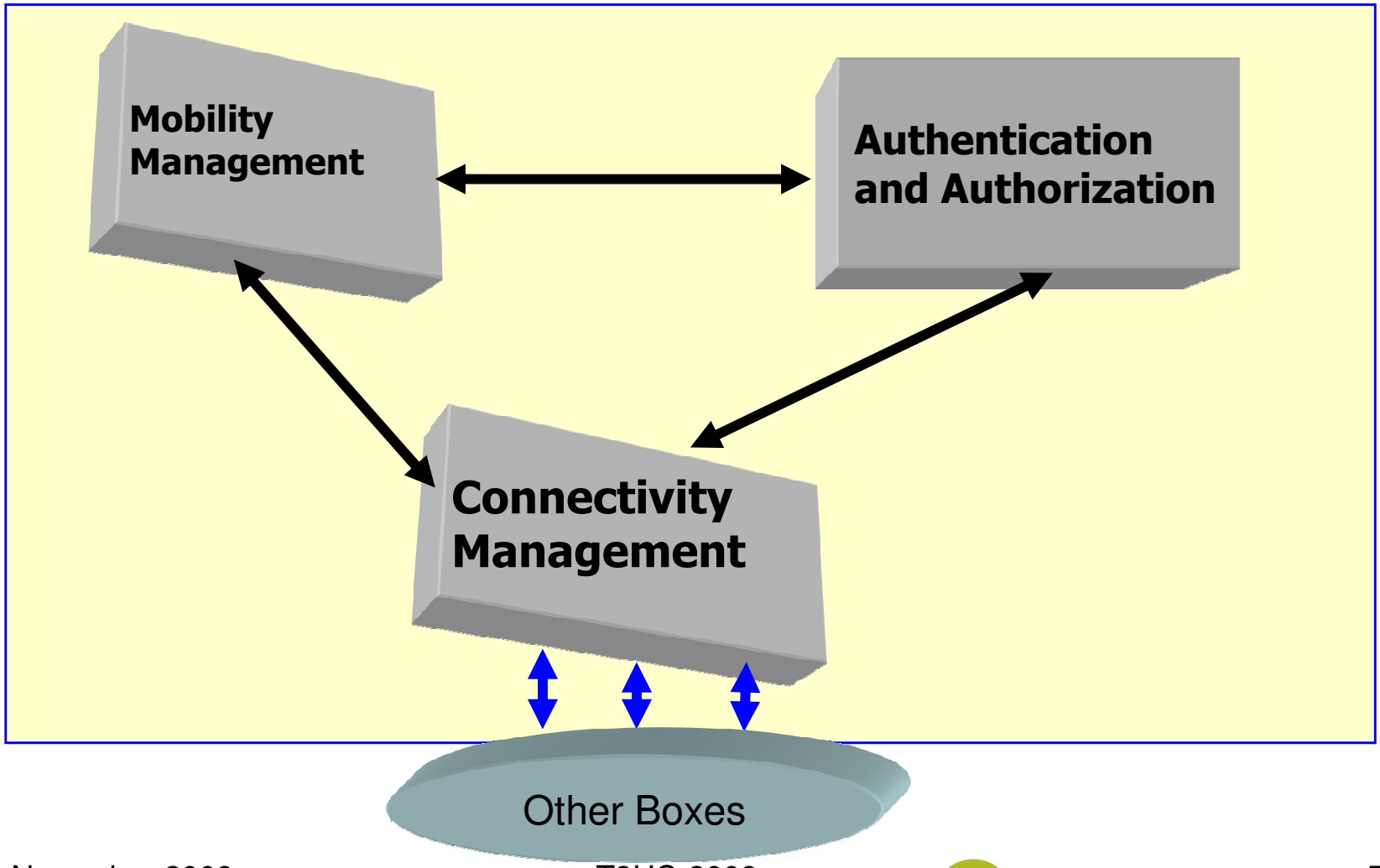
WiMAX Reference Architecture



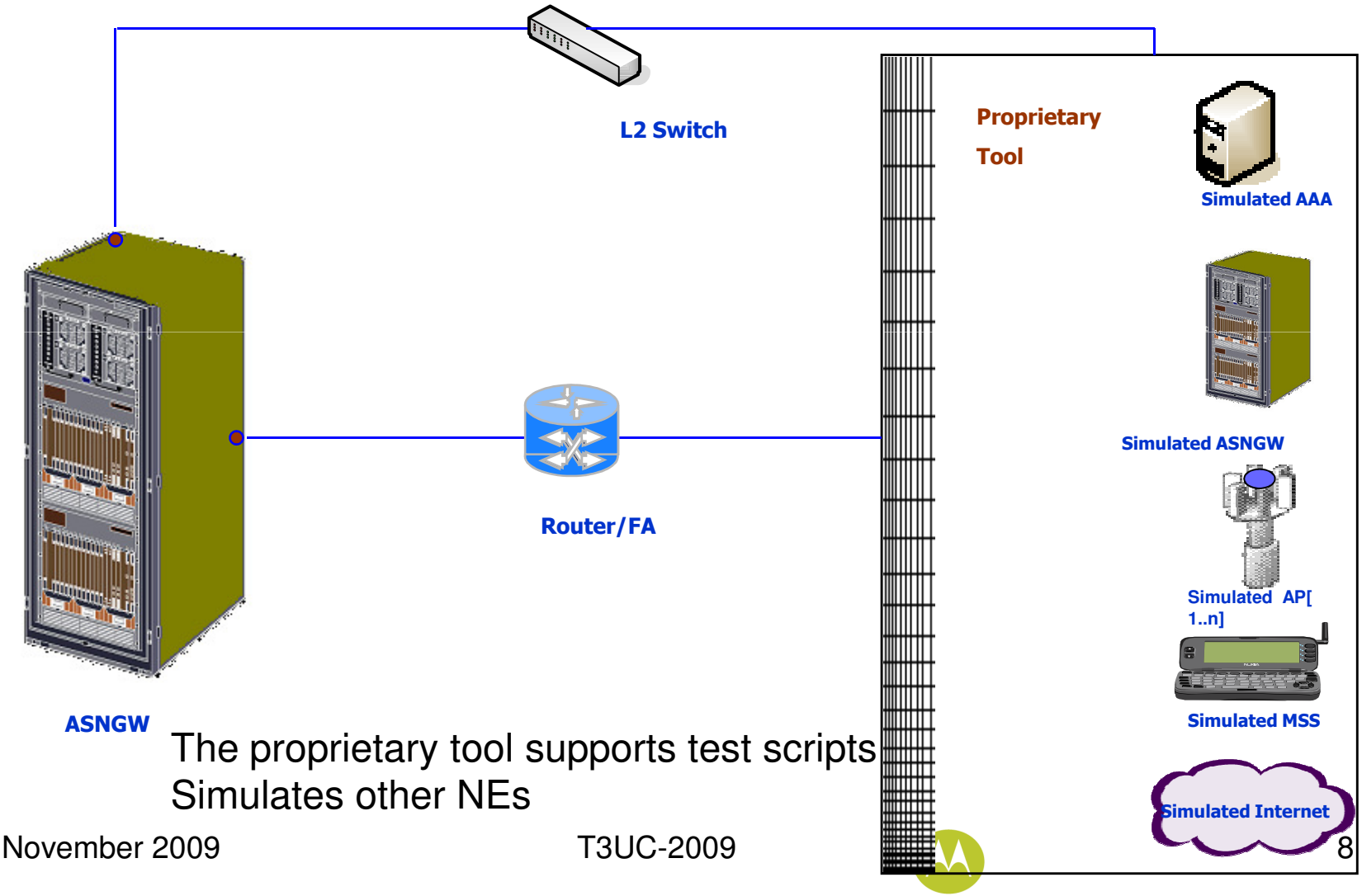
ASNGW: Decision Point

- Decision Point runs on ATCA chassis
- HA design
- Provides Authentication and Authorization for SS
- Mobility management for SS
- Connectivity management with Access Points, FA/Router and other neighboring ASNGWs
- Confirming to NWG standard

ASNGW Decision Point: Feature Architecture



WiMAX ASNGW Test Setup

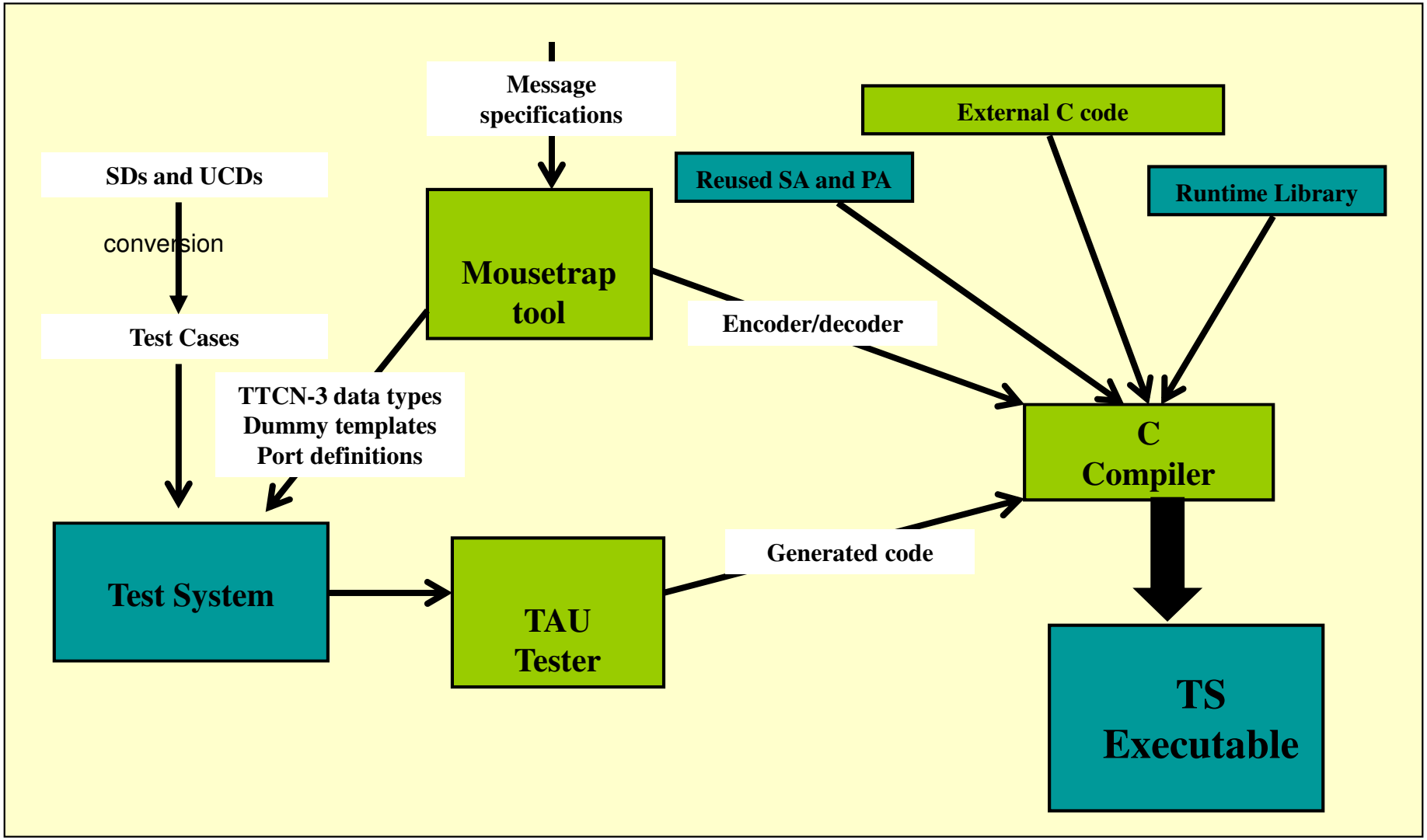


The proprietary tool supports test scripts
Simulates other NEs

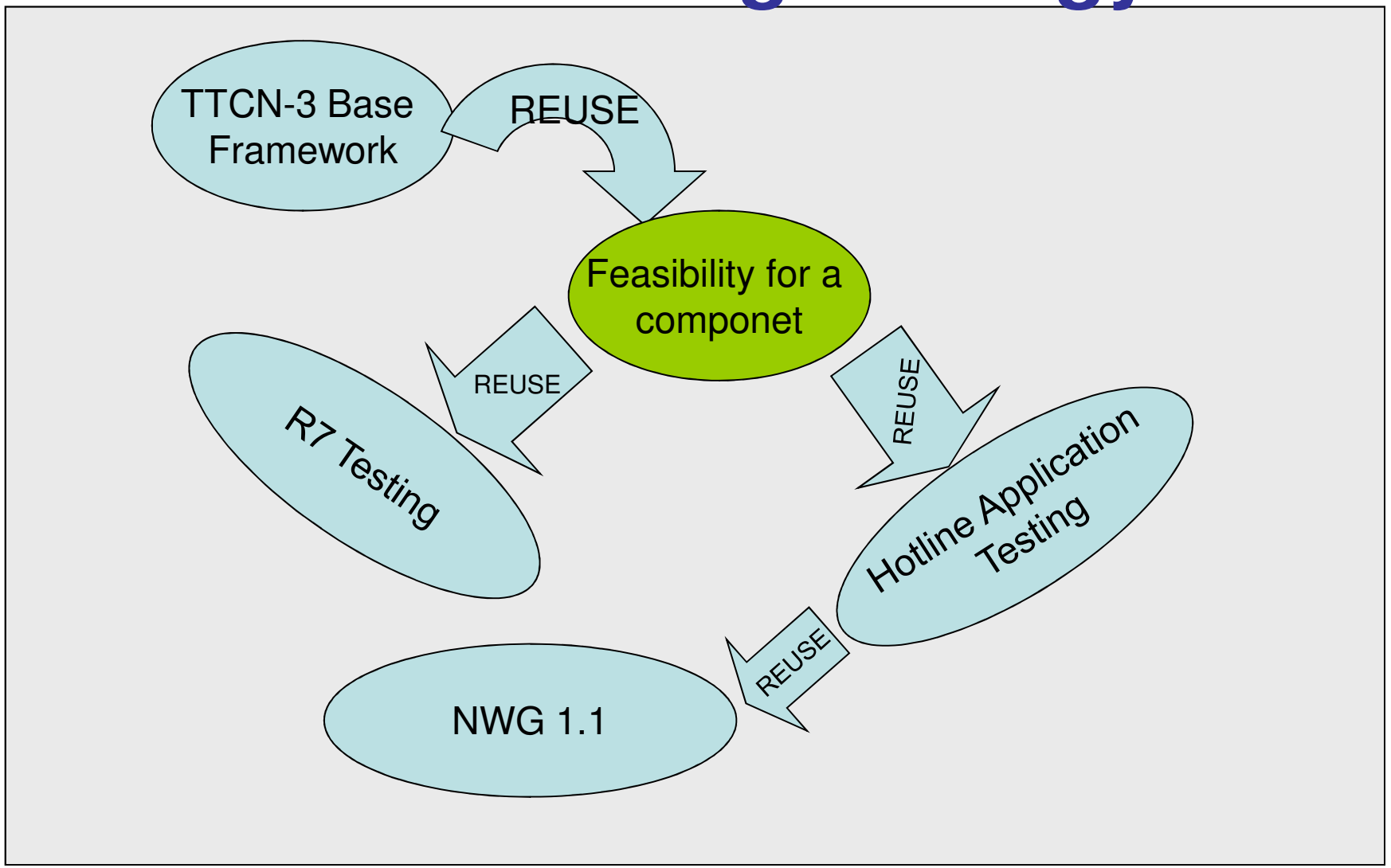
Motivation for TTCN-3

- Component testing and applicable beyond
- Standard language for test specification
- Rich Language features to meet testing needs of WiMAX features
- Tool support through –Tau G2 Tester
- Good automation support and ready made framework
- Regression capabilities
- Automated testing without need for re-compilation
- Verdict handling
- Automated logging with failure reasons
- Command line execution, compilation and generation
- Dynamic logging

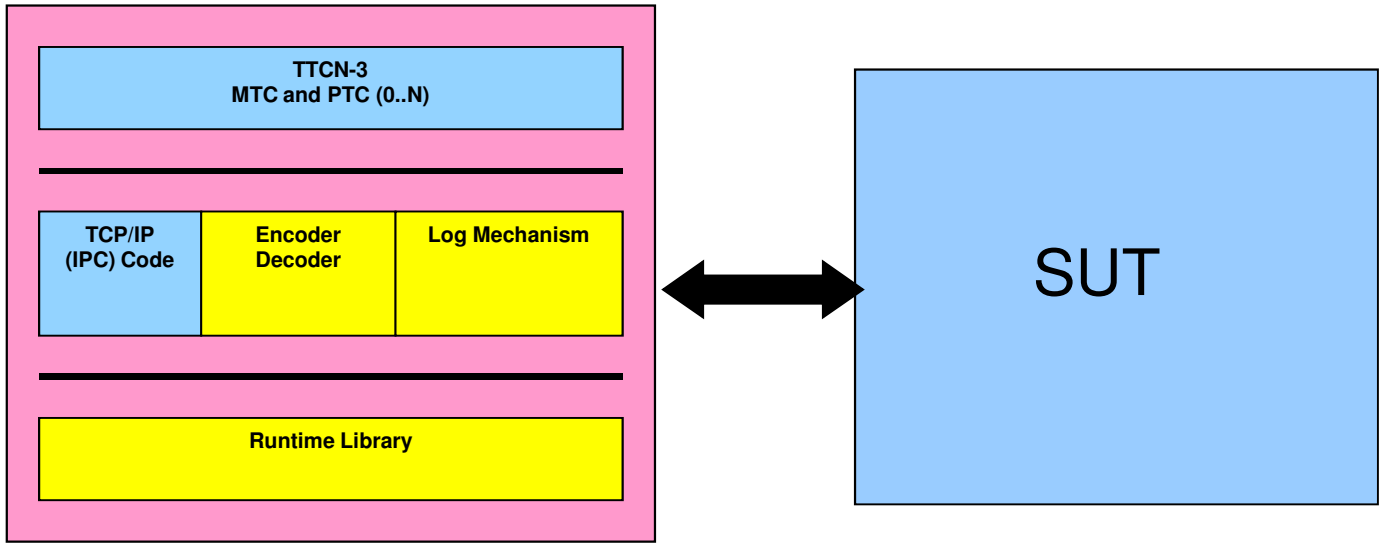
Automation framework



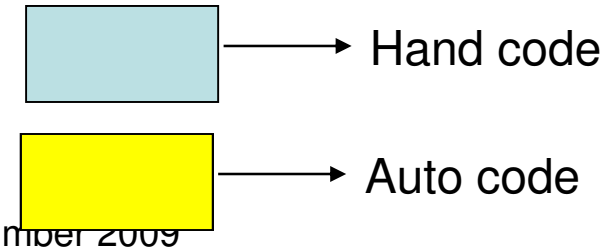
Feature Testing Strategy



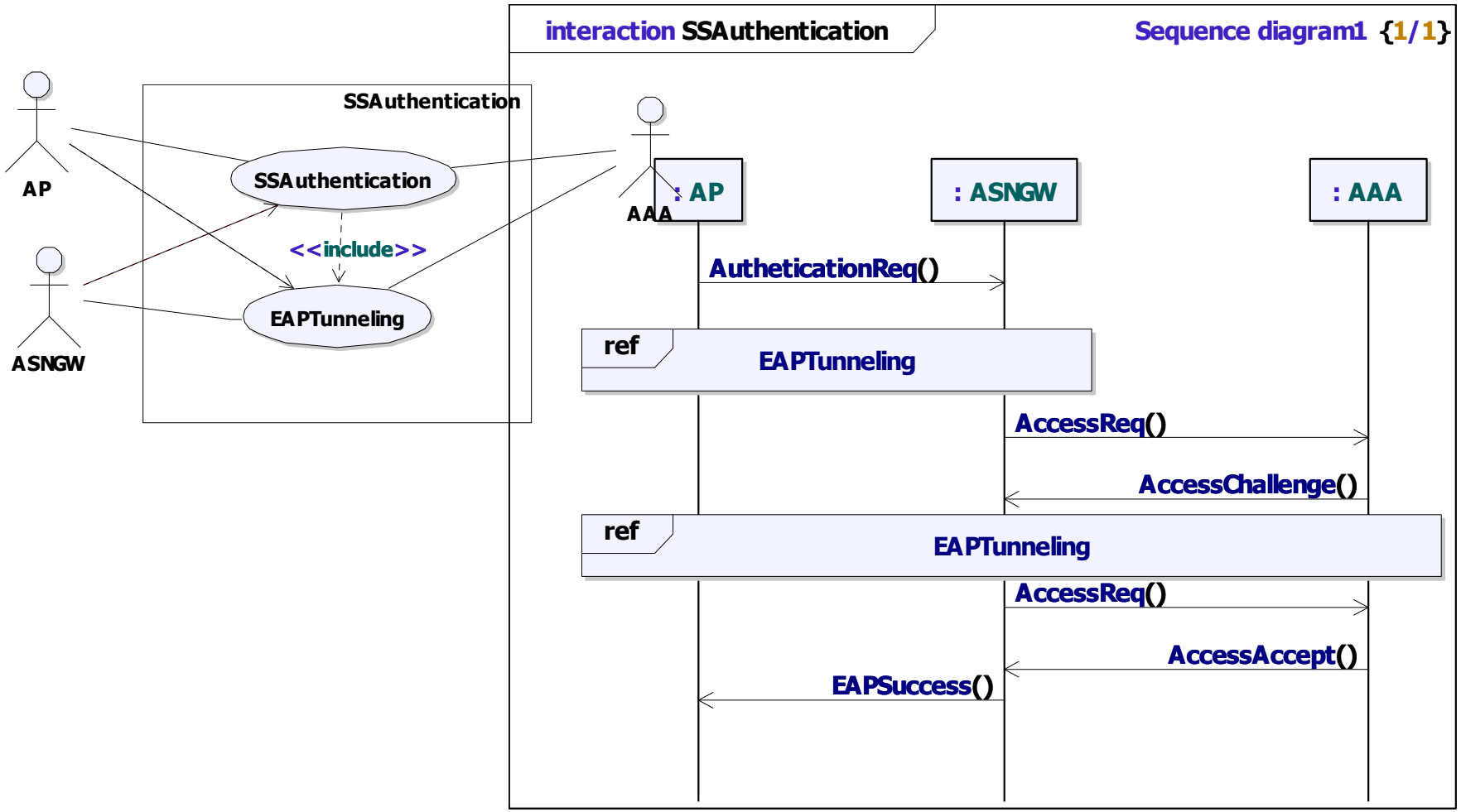
Test system architecture



Test System (TS)



Test Specification



Sample Test case

```
type component  MTCType
        {
                port  RADIUS_INTERFACE_type mtcPort_RADIUS_INTERFACE;
                port ApAsngw_type mtc_ap_asngw_port ;
                timer T;

        }
type component TSI
        {
                port  RADIUS_INTERFACE_type tsiPort_RADIUS_INTERFACE ;
                port  ApAsngw_type tsi_mtc_ap_asngw_port ;
        }
```

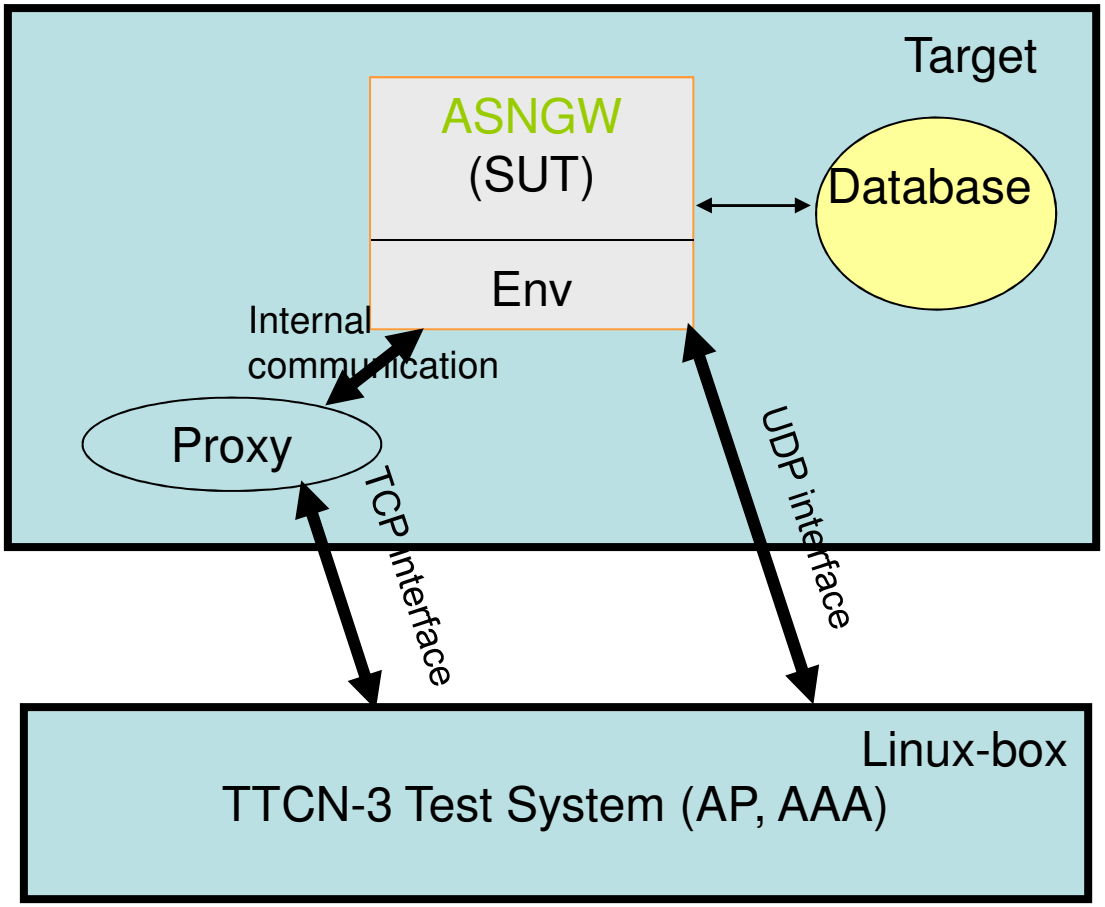
```
testcase TC1 (integer configParams) runs on MTCType system TSI
{
   //variables declaration
   // mapping of ports
      mtcPort_.send(AUTH_MyTemplate);
      alt {

           []mtcPort_auth_.receive(EAP_REQ_MyTemplate) -> value eap_req_var {

      mtcPort_auth_.send(EAP_RSP_MyTemplate_PAR(eap_req_var))
                                        alt {
.....

}
```

Test Execution set-up



Challenges faced

- SUT testing
- Health-check integration
- Integration of encryption algorithm
- Supporting various test configurations
- Communication aspects
  - Integration with proprietary paradigm
- Issues pertaining to integration of shared memory
  - Population of shared memory
- Issues pertaining to real-time nature of the application
  - When full log is enabled, reboots
  - Controlling of response timers

Benefits

- Standard language for testing
- Automation capabilities
- Reuse
  - Results of one feature testing have helped the usage and extension to the other features.
  - The test architecture and reuse of the test system across various features
- Ability of TTCN-3 to handle configuration, execution and tracing of test case execution
- Better cycle time, quality and productivity

Take-away

- Organizations which are not using TTCN3
  - Technology is good for any message based testing
  - For API based testing, need to investigate synchronous communication
- Organizations which are using TTCN3
  - Development of common TTCN-3 framework
  - Automation of test system components
- Research organizations and tool vendors
  - Automation support may be provided by the vendors
    - Automatic generation of encoders/decoders, templates, communication parts
    - Thoughts on …Domain specific common test framework

Conclusions

- TTCN-3 approach is flexible to adopt to various features and testing of components independently
- Automation of TTCN-3 test system components
- Reuse of Message specifications across SUT and TS
- Reuse of framework across test phases
- Good cycle time reduction to market, higher quality and greater productivity
- Challenges will be there to address
  - Ex: Encryption algorithms etc

Q&A