Standards for Business

Interoperability Testing in ETSI

Steve Randall

PQM Consultants, UK

Vice-Chair, ETSI TC-MTS

2nd ITU-T Informal Workshop on
Conformance and Interoperability Testing

Structure of this Presentation

- □ Background
- □ ETSI’s Interoperability Methodology:
  - ➢ Interoperability and Conformance
  - ➢ Basic concepts
  - ➢ Developing Interoperability Tests

Background

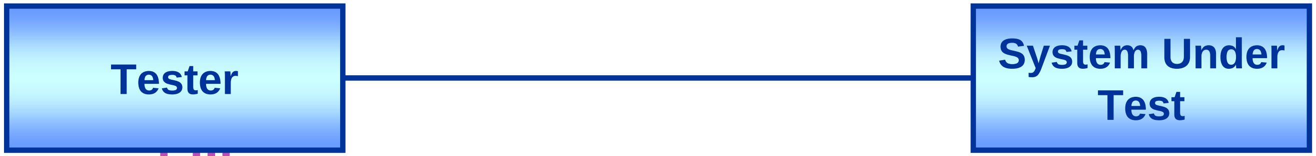
- □ Telecommunications standards bodies have traditionally specified only conformance tests for their protocols and services;
- □ Interoperability testing has been considered to be a purely commercial issue;
- □ ISO 9646 is still a relevant methodology for conformance testing purpose;
- □ Pressure from committees within ETSI has resulted in TC-MTS developing an “ISO 9646 for Interoperability”

ETSI’s Interoperability Testing Methodology

- ❑ Originally published as a project-specific methodology (TS 102 237-1)
- ❑ Draft generic methodology currently out for committee approval with publication expected March 2007
- ❑ Methodology covers two main segments:
  - ➢ Interoperability test development;
  - ➢ Interoperability testing.
- ❑ The testing methodology is only one small part of ETSI’s overall approach to interoperability

Interoperability and Conformance - 1

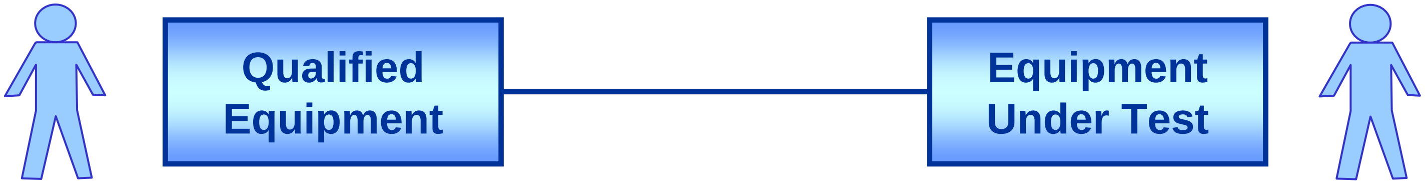
- **Conformance**
  - **Establishes whether or not the implementation of a standard meets the requirements specified in the standard.**



Tester

System Under Test

- **Interoperability**
  - **Assesses the ability of an implementation to support the required functionality between itself and another implementation.**



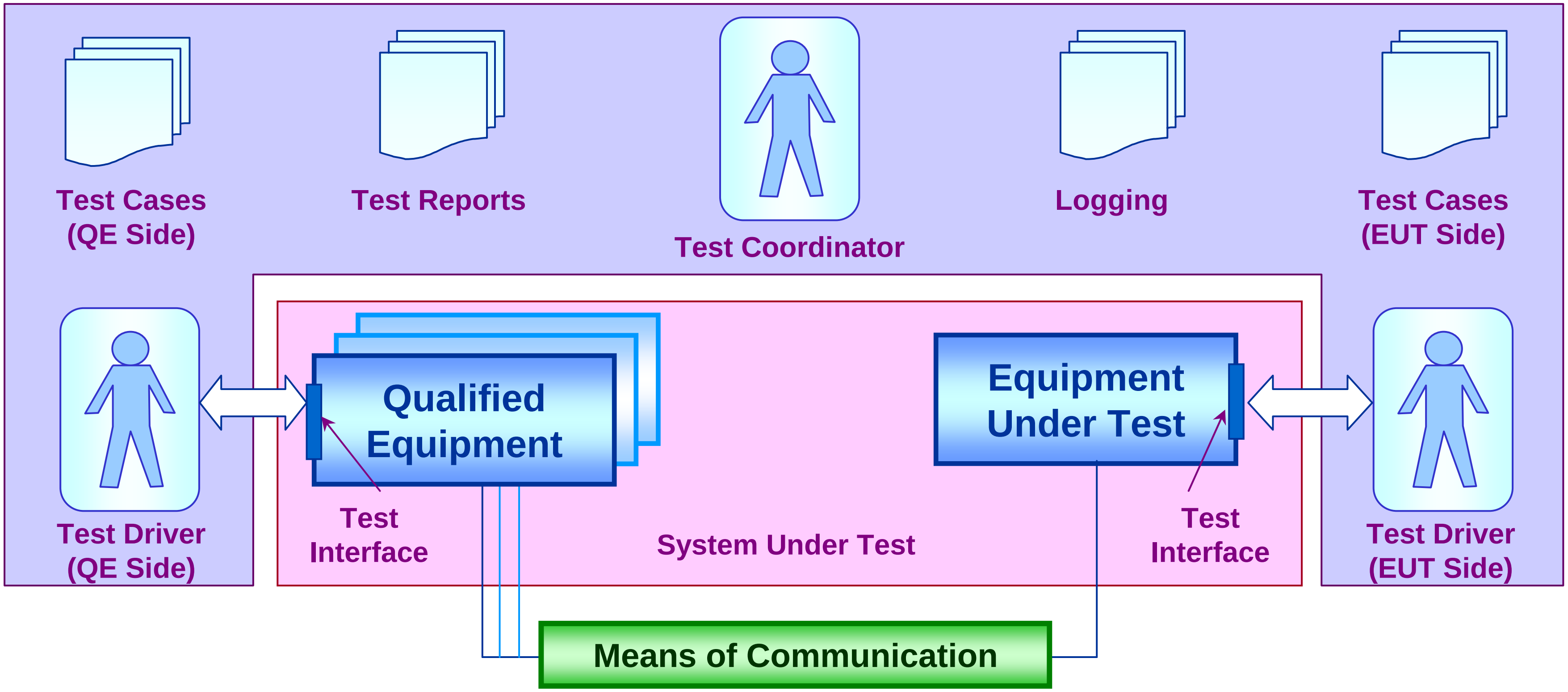
Qualified Equipment

Equipment Under Test

Interoperability and Conformance – 2

- Interoperability testing is not a substitute for conformance testing
- Conformance testing does not assure interoperability
- Conformance testing together with interoperability testing is a very powerful combination

Basic Interoperability Concepts
1 – General Test Arrangement



Basic Interoperability Concepts 2 –Principal Components

- ❑ Equipment Under Test (EUT)
  - ➢ Only one in any SUT
  - ➢ Hardware, Software or a Combination
  - ➢ Not previously tested successfully
- ❑ Qualified Equipment
  - ➢ Any number in an SUT
  - ➢ Hardware, Software or a Combination
  - ➢ Previously tested successfully
- ❑ Test Driver
  - ➢ Human or machine controller of EUT and QE(s)
- ❑ Test Coordinator
  - ➢ Human or machine controller and synchronizer of tests
  - ➢ May also be one of the Test Drivers

Interoperability Test Specification

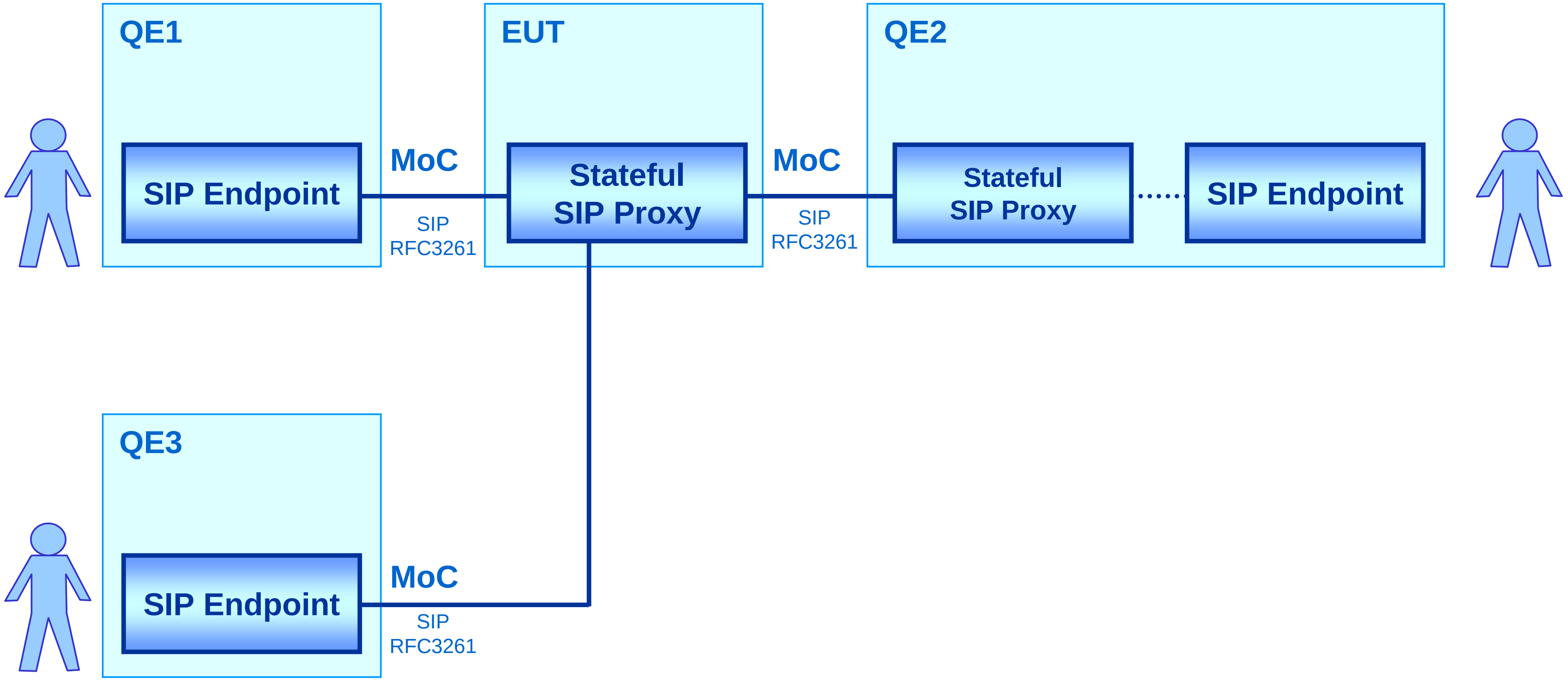
- ❑ **Specification process steps**
  - ➢ Specify abstract architecture
  - ➢ Write draft Interoperable Features Statement (IFS)
  - ➢ Specify Test Suite Structure (TSS)
  - ➢ Write Test Purposes (TP)
  - ➢ Write Test Cases (TC)
  - ➢ Validate Test Cases

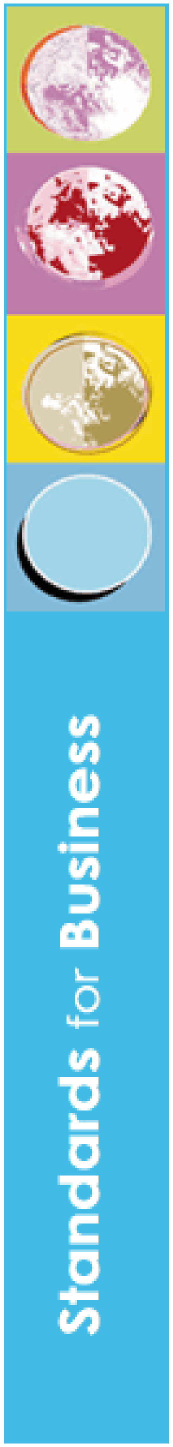
Standards for Business

Specify Abstract Architectures

- ❑ General framework(s) within which concrete test arrangements must fit
- ❑ Can be graphical, tabular or textual
- ❑ Should identify:
  - ➢ EUT
  - ➢ QE(s)
  - ➢ Paths between EUT and QE(s) (MoC)
  - ➢ Valid equipment types for EUT and QE(s)
  - ➢ Expected protocols to be used

Example Abstract Architecture





Write Draft Interoperable Functions Statement

- ❑ An IFS Identifies:
  - ➢ Functions that an EUT must support
  - ➢ Functions that are optional
  - ➢ Functions which are conditional
- ❑ The IFS provides structure to the test specification
- ❑ It can also be used like a PICS as a proforma for a manufacturer to declare which functions are supported in an EUT

Specify Test Suite Structure

Identify test groups based upon, e.g.:

- ➢ Abstract Architecture
- ➢ Functionality
- ➢ Behaviour:
  - Normal
  - Exceptional

❑ Define test coverage for each group

- ➢ What range of tests is to be included in each test group

Write Test Purposes

- ❑ For each possible test case, describe WHAT is to be tested
- ❑ Use the most appropriate means of expressing Test Purposes:
  - ➢ Plain language
  - ➢ Tables
  - ➢ MSCs
  - ➢ A specialist notation such as TPLan which offers:
    - Consistency in TP descriptions
    - Clear identification of preconditions, test actions and verdict criteria
    - Checkable syntax

Write Test Cases

- ❑ Test cases should include:
  - ➢ Preconditions
    - Configuration
    - Initial status
  - ➢ Test steps
    - Detailed instructions to Test Driver
      - Clear
      - Precise
      - No unnecessary restrictions
  - ➢ Verdicts
    - “Pass” means “EUT Pass”!
    - “Fail” may not mean” EUT Failure”
      - QE failure
      - MoC failure
      - Requires investigation

Test Case Specification

- Tabulated free text
  - Ideal for implementation by human Test Drivers
  - Individual test steps and their relation to each other is easy to understand
  - Only supports simple, serial test path, .i.e, very difficult to describe alternate paths following an unsuccessful intermediate verdict
- Programming script or language (e.g., PERL, TTCN-3)
  - Ideal for machine implementation of Test Drivers
  - Highly repeatable
  - Allows comprehensive handling of unexpected behaviour
  - Difficult for the human user to read and follow
  - Establishing a testing environment is complex

Sample Test Case Tabular

Identifier	TC_SS_0001_01		
Summary:	Supervised call transfer from User B to User A		

Test Purpose:	ensure that {
	when { A call is established between User_C and User_B }
	then { User_B can transfer the call from User_B to User_A
	after User_B and User_A communicate }
	}		

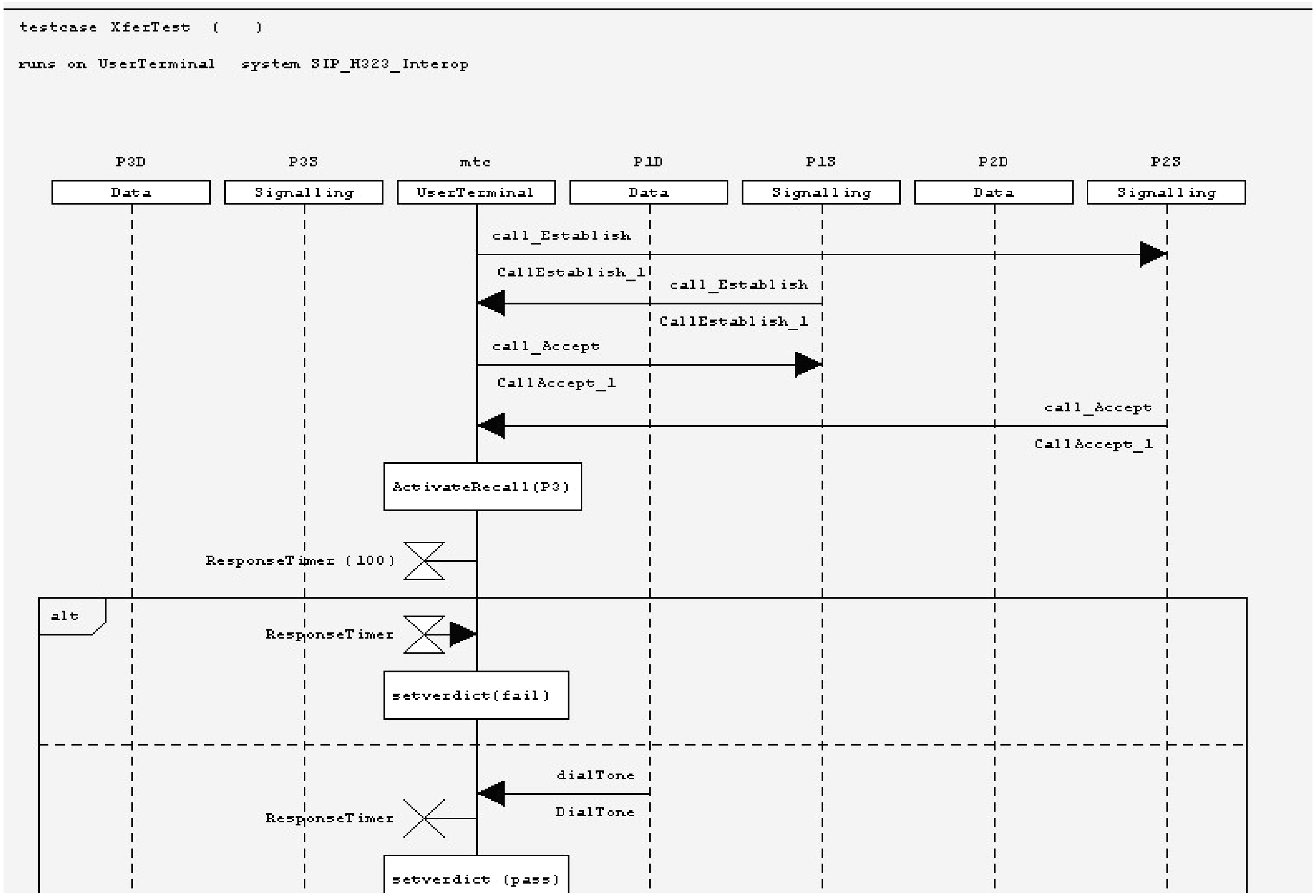
TP Identifier	TP_SS_0001	**Configuration:**	Test Architecture 2
Pre-test conditions:	• User A, User B and User C configured with Bearer Capability set to "Speech, 64 kbit/s" • User A configured to support the Call Transfer service		

Step	**Test sequence**	**Verdict**	
		Pass	**Fail**
1	Initiate new call at User C to the address of User B		
2	Accept call at User B		
3	Activate the "recall" button (or equivalent) at User B's terminal		
4	*Is dial tone (or an equivalent indication) present at User B's terminal?*	Yes	No
5	Initiate a new call from User B to the address of User A		
6	*Is User A's terminal alerting (visual or audible indication)?*	Yes	No
7	Accept call at User A		
8	Apply speech at User A		
9	*Can speech from User A be heard and understood at User B?*	Yes	No
10	*Can speech from User A be heard and understood at User C?*	No	Yes

Sample Test Case
TTCN-3 (Core)

```
// Define Supervised Transfer test case
testcase SupervisedTransfer() runs on userTerminalType
{  timer ResponseTimer := 100E-3;

  // Preamble: Establish call between Users B & C
  m3s.send (CallEstablish_1);
  m2s.receive (CallEstablish_1);
  m2s.send (CallAccept_1);
  m3s.receive (CallAccept_1);
  // Register recall test
  m2s.send (Recall);
  ResponseTimer.start;
  alt
  { [] ResponseTimer.timeout
       { setverdict(fail);
         stop
       }
    [] m2d.receive (DialTone)
       { setverdict(pass);
         ResponseTimer.stop
         // Hold call test
. . . . . . .
```

Sample Test Case
TTCN-3 (GFT)



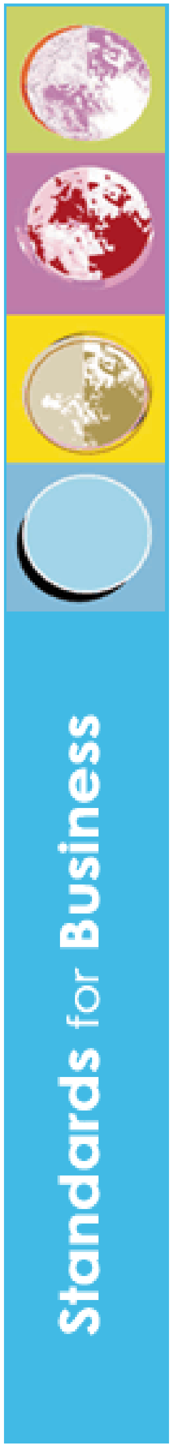
```
testcase XferTest (    )

runs on UserTerminal   system SIP_H323_Interop
```

Validate Test Cases

- Ideally, test cases should be validated in a lab
- Structured Walkthrough/Peer Review is the minimum level of validation acceptable
- Validation ensures that:
  - Pre-conditions make the SUT ready for testing
  - No unnecessary pre-conditions are specified
  - Abstract architecture can be realized in an appropriate concrete configuration
  - Test steps are unambiguous and easy to follow
  - Each test case realizes objective of its test purpose
  - Combined verdicts give true assessment of Test Purpose

And finally . .

- ETSI's initial Interoperability Test Methodology:
  - Published in 2003 (TS 102 237-1)
  - Project specific
  - ISO 9646 principals
  - Used in a number interoperability testing projects
- Revised Interoperability Test Methodology:
  - To be published in 2007
  - Generic to any project
  - Only minor changes on the basis of experience
- Interoperability testing is important, BUT . . . . . .
- "Design for Interoperability" is more important!



Thank You!

Interoperability Testing in ETSI