Specification-Based Testing of Communication Protocols

Adam Wolisz, Ina Schieferdecker

Technical Univeristy Berlin and GMD Fokus

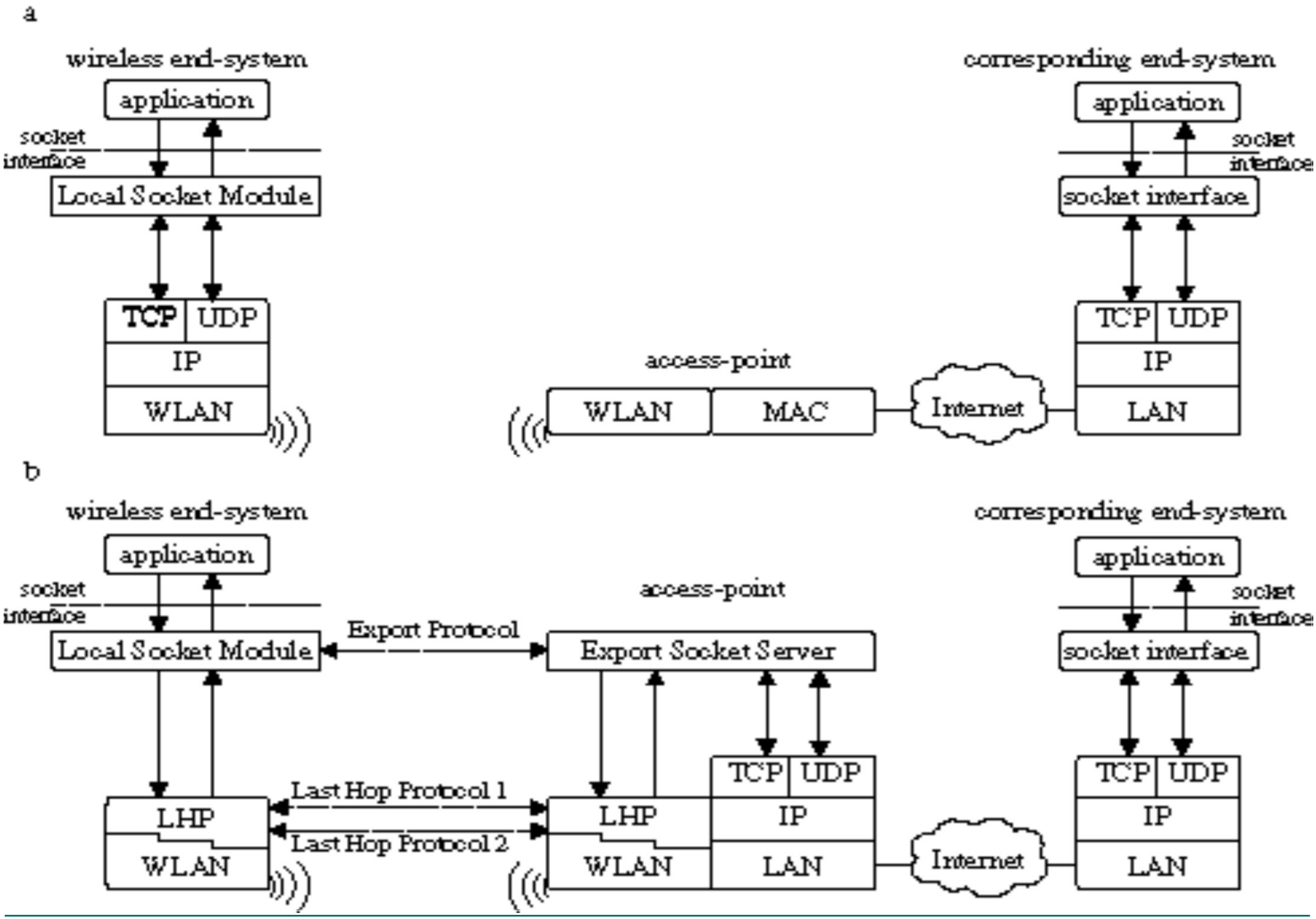
www-tkn.ee.tu-berlin.de
www.fokus.gmd.de/TIP

Contents

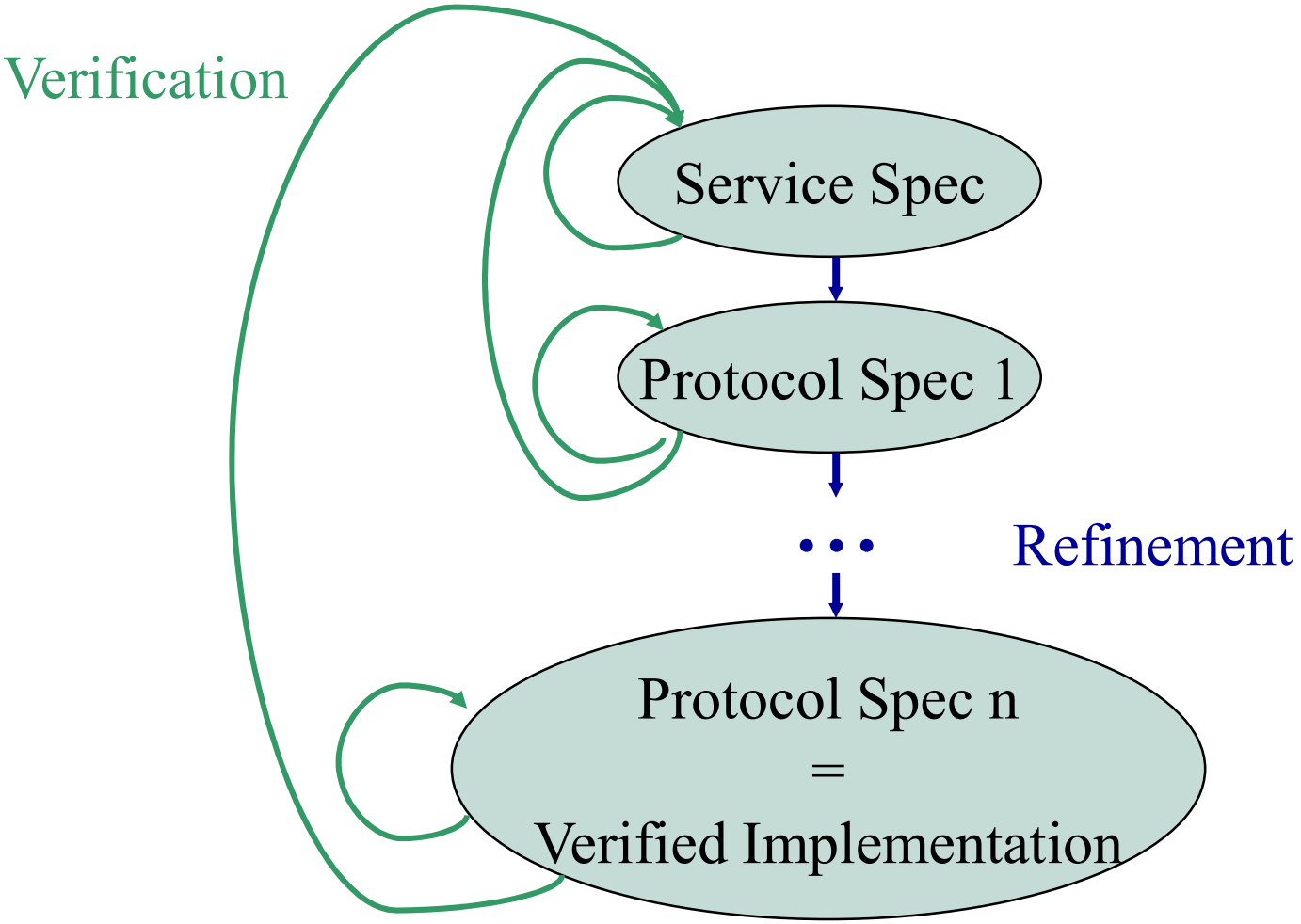
- System Development.
- Conformance/Performance Testing.
- Testing for Wireless Communication.
- Test Specification.
- Conclusions.

Services vs. Protocols

- Service specification is what matters! (API, API, API)
- Several protocols (open or proprietary) can realize a given service and assure the desired service semantics.
- A **subset** of a heavy-weight standardized protocol behavior might be sufficient for supporting a service.
- Example 1: Wireless Profibus (a variant of Fieldbus)
- Example 2: Remote Socket Architecture
  - Supports end-to-end **socket** semantics (not full TCP)
  - Combined use of cascaded protocols.


a
wireless end-system
application
socket interface
Local Socket Module
TCP
UDP
IP
WLAN
corresponding end-system
application
socket interface
socket interface
TCP
UDP
IP
LAN
access-point
WLAN
MAC
Internet
b
wireless end-system
application
socket interface
Local Socket Module
Export Protocol
access-point
Export Socket Server
Last Hop Protocol 1
Last Hop Protocol 2
LHP
WLAN
TCP
UDP
IP
LAN
corresponding end-system
application
socket interface
socket interface
TCP
UDP
IP
LAN
Internet

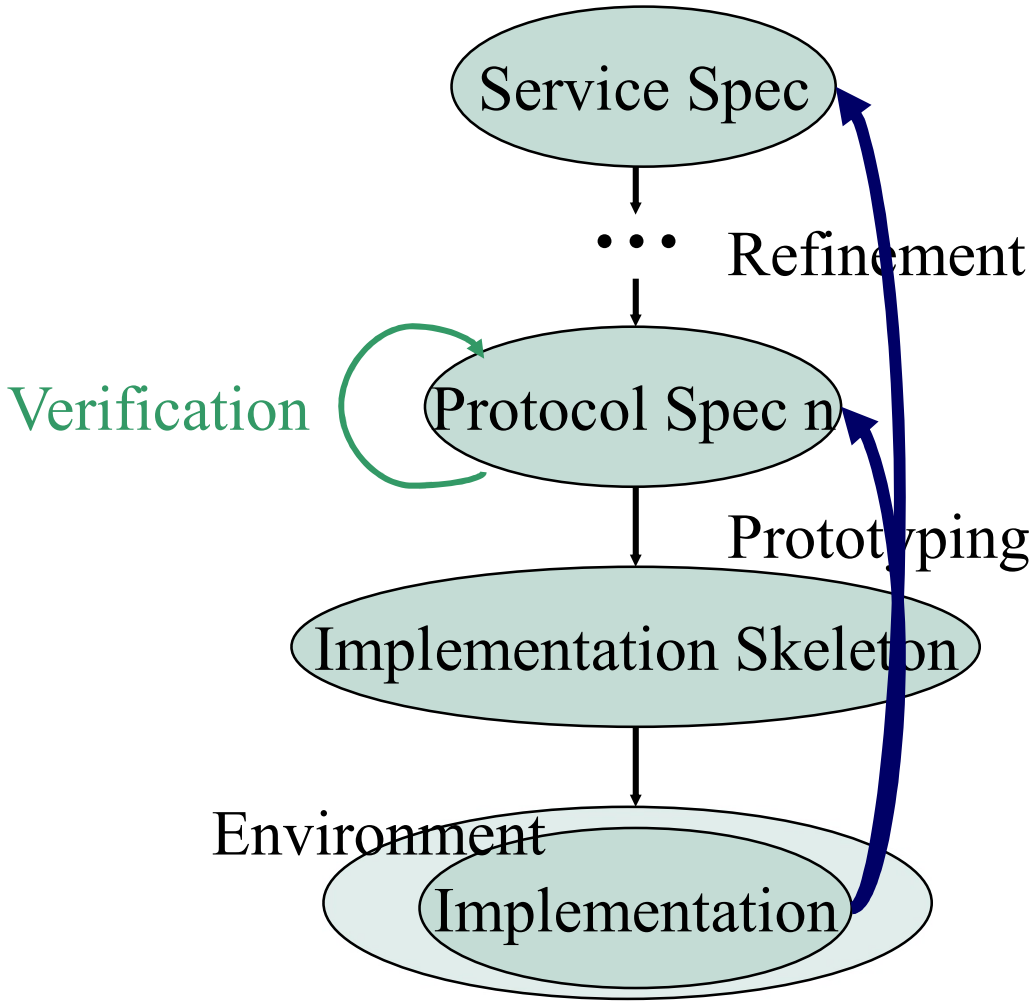
The Ideal Picture



However ...

- Typically, the protocol is not verified w.r.t. the service.
- Specifications abstract from implementation details
- implementation skeletons only.
- The target environment is not specified and not verified
- eventually installed protocol is not verified.

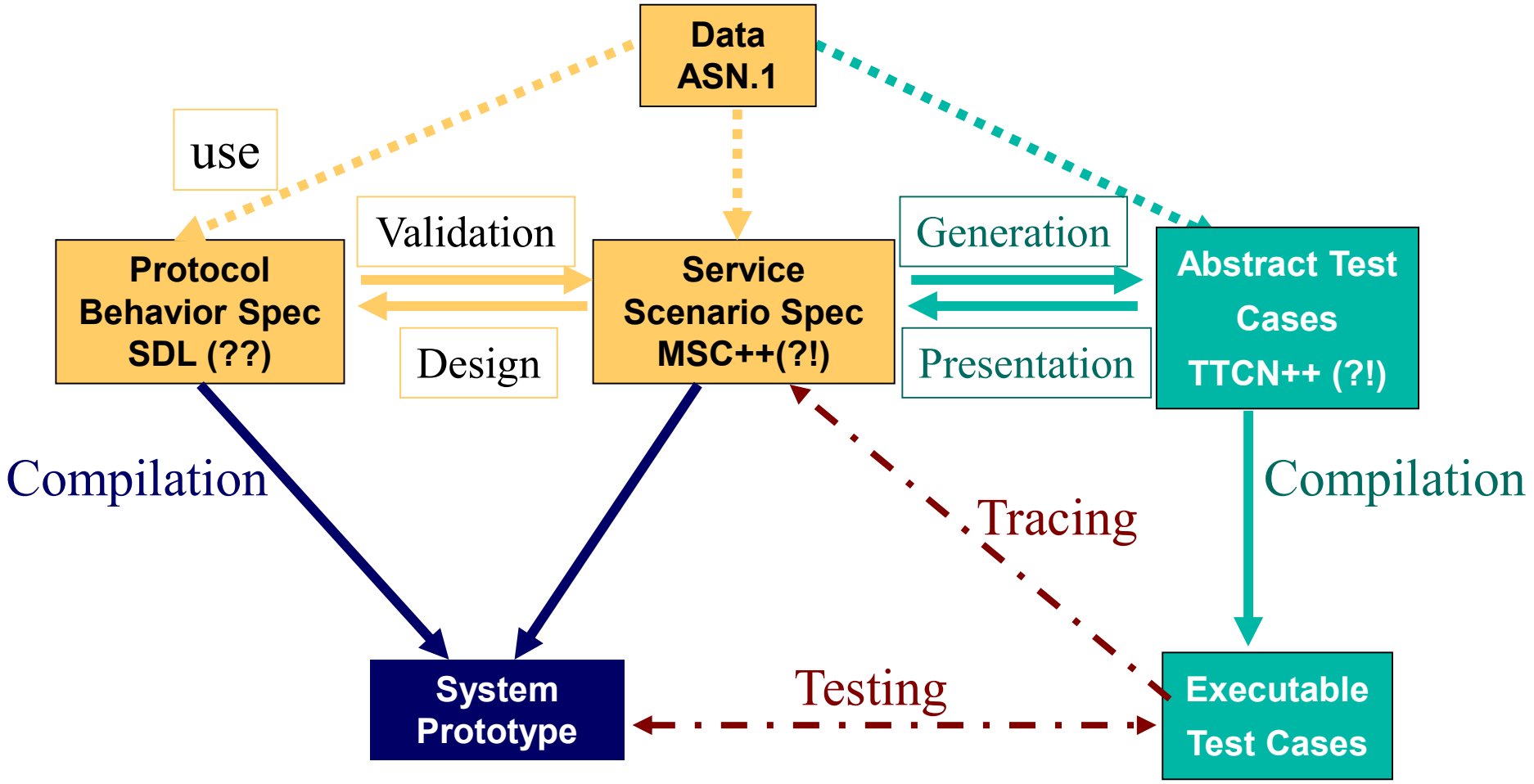
The Real Picture



Service Testing ?

Protocol Testing ?

Specification-Based System Development



Testing Objectives

- Testing is the process of verifying that an implementation performs in accordance with a particular specification.
- Testing is performed in order to find errors if existent.
- Testing approves a quality level of a tested system.

However:

- Testing cannot guarantee the absence of errors in an implementation; because,
- exhaustive testing is not practical and too expensive in most cases.

Test Ingredients

- Target of testing: implementation under test (IUT) within system under test (SUT).
- Means of testing: (system of) test device(s).
- Test assumptions: a priori taken.
- Test oracle: test results deduced from observation.

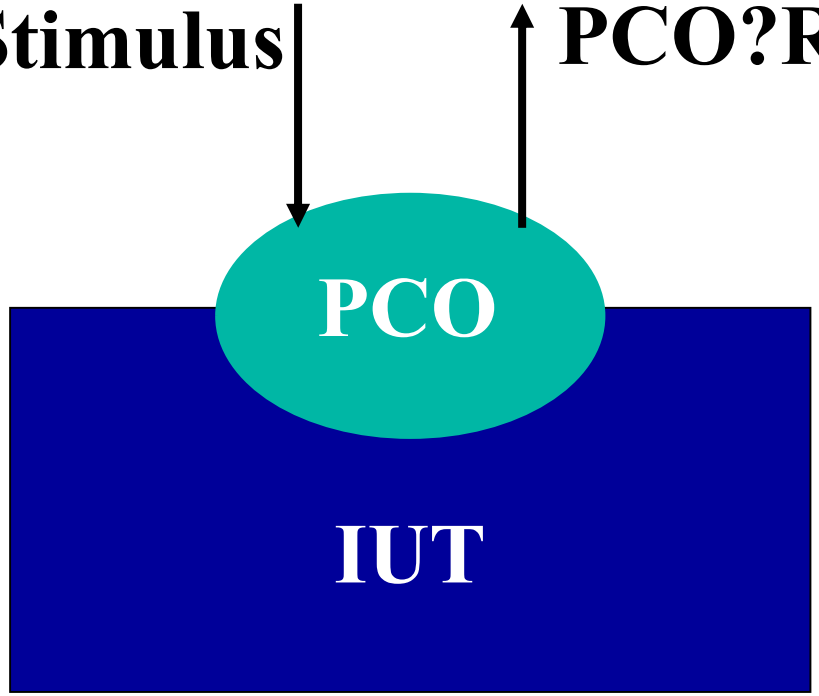
- Optimization process: test assumptions vs. test oracle.
- Goal: relevant,**reproducible**,accurate test results.

Testing Approaches

- Black-box testing
  - Behavioural tests based on requirements derived from the specification of the protocol or service.
- White-box or glass-box testing
  - Structural tests derived from the structure of the tested object (i.e., testing the program structure).
- Grey-box testing
  - Hybrid testing approach.

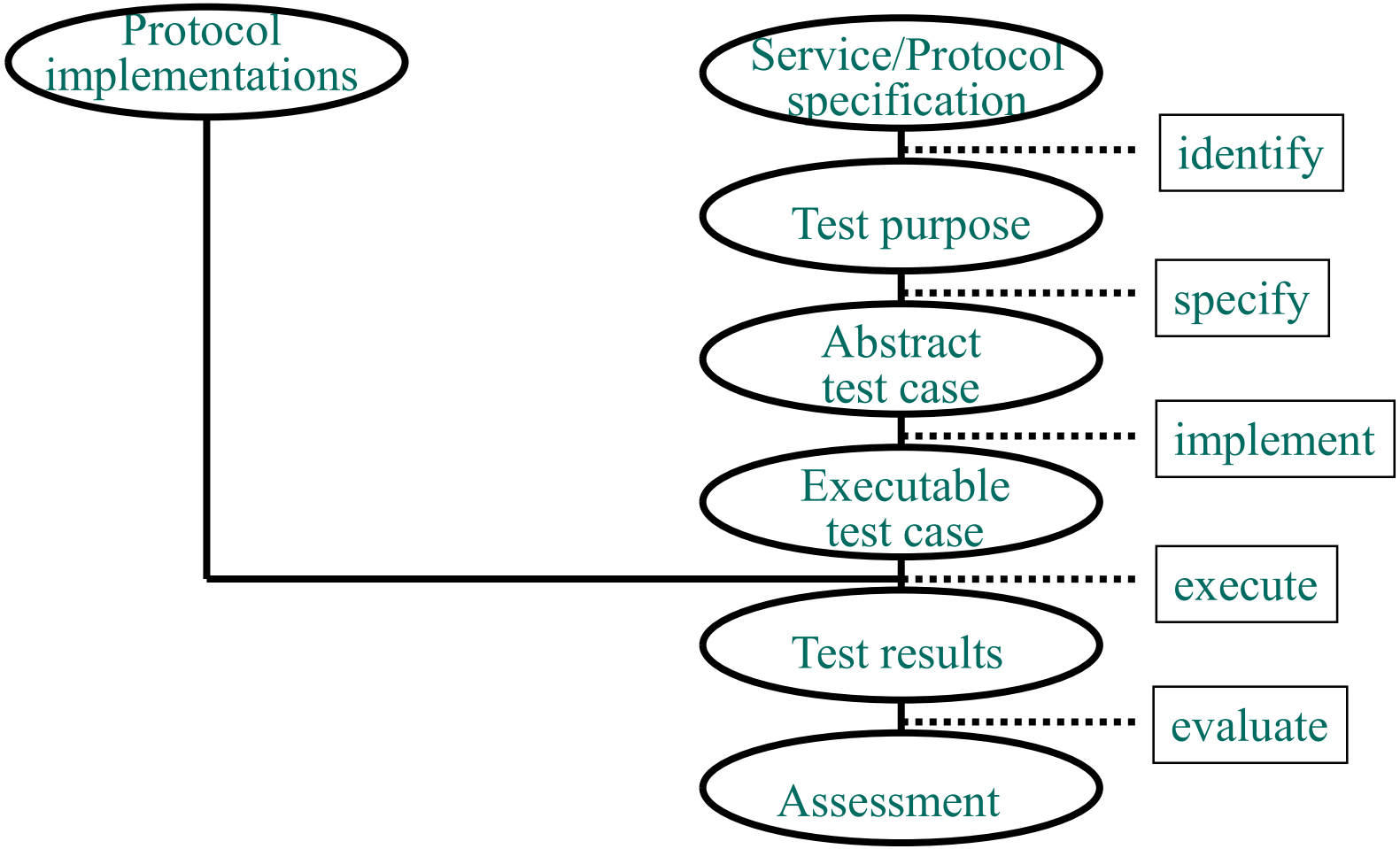
PCO Model for Black-Box Testing

- Stimulate IUT.
- Compare observed response with foreseen response.
- Assign verdict.



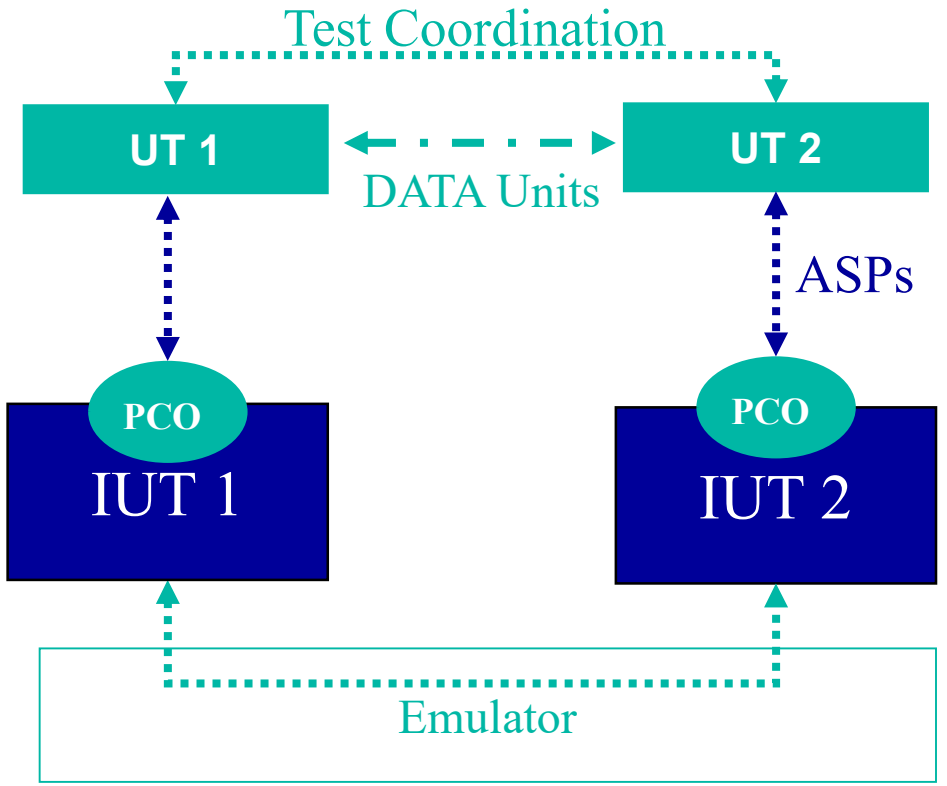
IUT: Implementation Under Test
PCO: Point of Control and Observation

How to Test?


Protocol implementations
Service/Protocol specification
identify
Test purpose
specify
Abstract test case
implement
Executable test case
execute
Test results
evaluate
Assessment

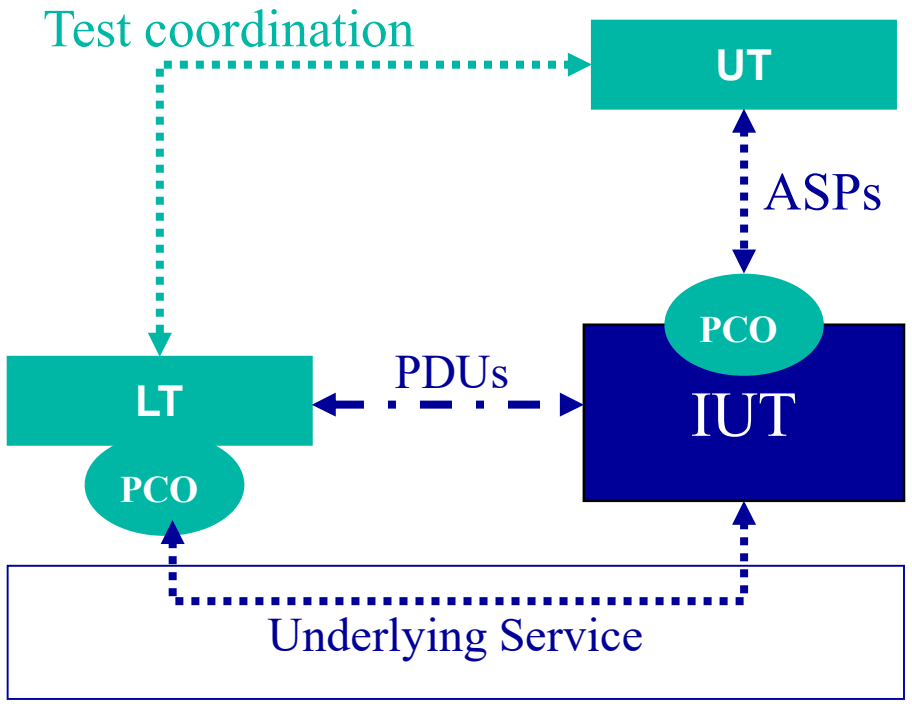
Service Test Configuration

- Access to upper boundaries of the protocol:
  Upper Tester (UT) with PCO
- Emulator might be complex (if needed with monitoring)



Protocol Test Configuration

- Access to upper and lower boundaries of the protocol:
  Lower and upper tester with PCO
- If needed, underlying service has to be emulated



UT: Upper Tester
LT: Lower Tester

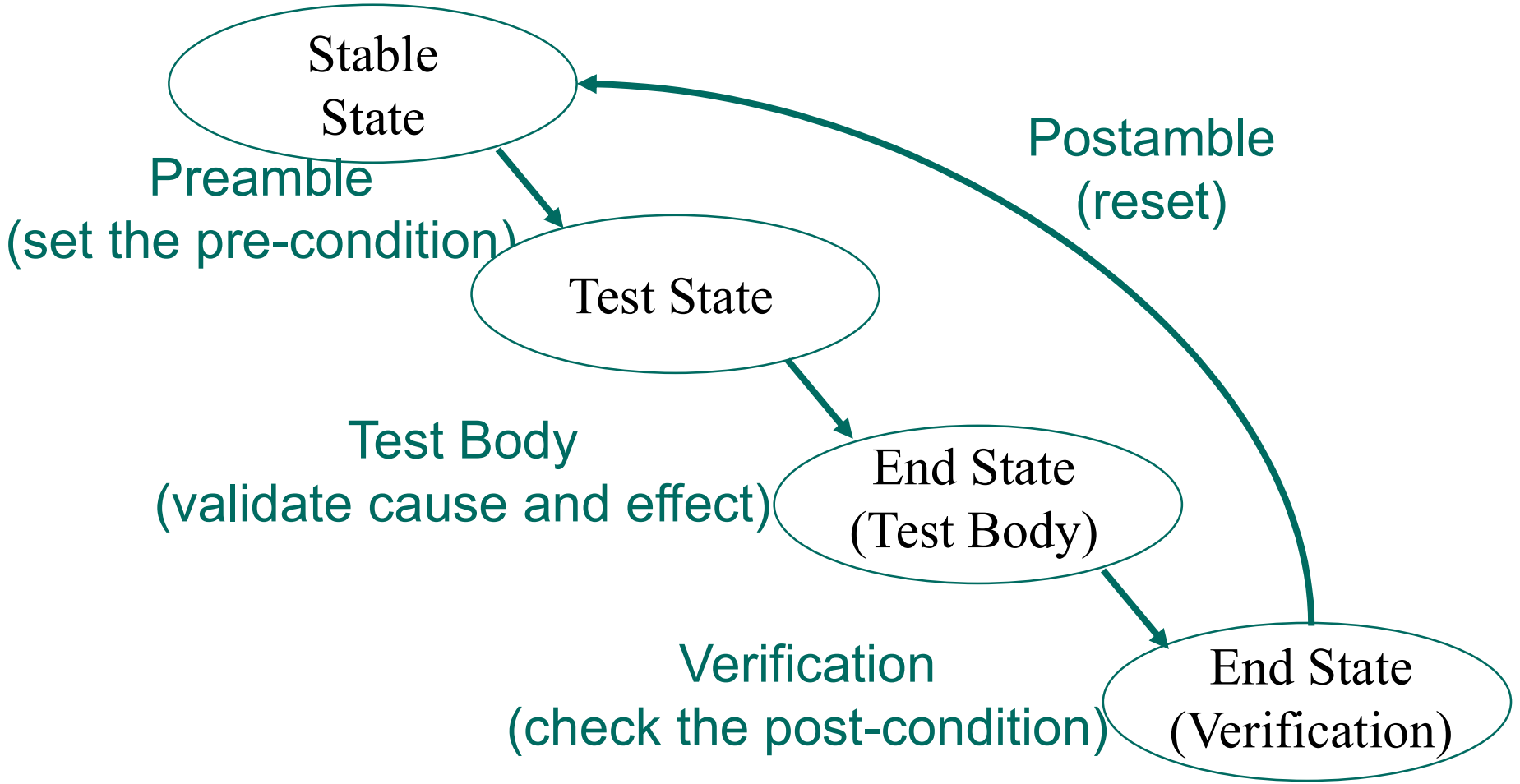
Test Sychronization Protocol

(ETSI - TSP1)

- Defines Procedures for
  - Test Setup,
    - Setting Up Configuration
    - Distributing Parameters, etc.
  - Test Execution
    - Start, Stop and Cancel
    - Routing of Coordination Messages
  - Test Reporting

- This is a distributed system...

Functional Test Case



State Transition Testing

- Model: **states** (disjoint, identifiable and finite in number), **transitions**, **events**, **actions**, **outputs:** pretty classical (or non-classical!) FSMs.
  - Test cases shall be designed to exercise transitions and states
- Some approaches support the notion of time, very few some notion of resource usage (limited parallelism of execution)
- No one - up to our info - supports the usage of energy.

Data Specification

- Data types
- Data values
- Data constraints
  - as a set of data values
  - can be seen as subtypes
- Data parameterization

- Encoding rules

Message Format Testing

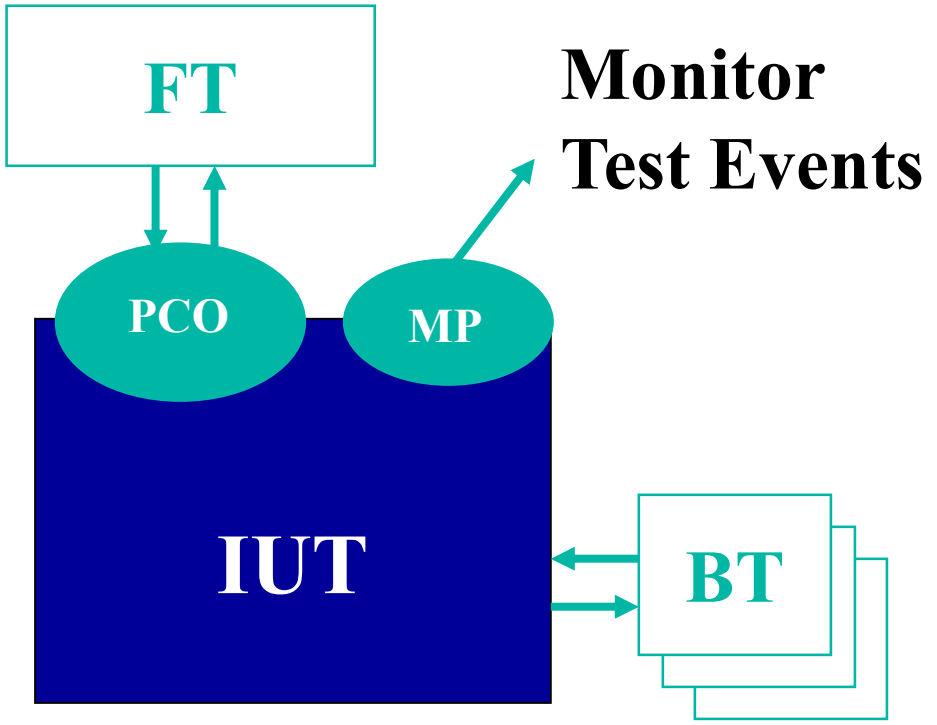
- Model of the formally-defined syntax of the inputs to the IUT:
  - Syntax as a number of rules each of which defines the possible means of production of a symbol in terms of sequences of, iterations of, or selections between other symbols.
- Test cases with valid and invalid syntax are designed.

Boundary Value Analysis

- Partition: an ordered set or range of values, which can reasonably be expected to be treated in the same way (i.e. considered ‘equivalent’). Identify boundaries!
- Valid and invalid values are partitioned
- Test cases shall exercise partitions.
- Typically, test cases corresponding to values on the boundary and an incremental distance either side of it.

Performance Testing

- Fore/background traffic with load patterns according to traffic models
- Monitoring of test events and their time stamps to measure the delay and rates of test events
- Statistical assessment of performance requirements and measured performance.
- Dream: be able to determine the most critical data configuration for some set..

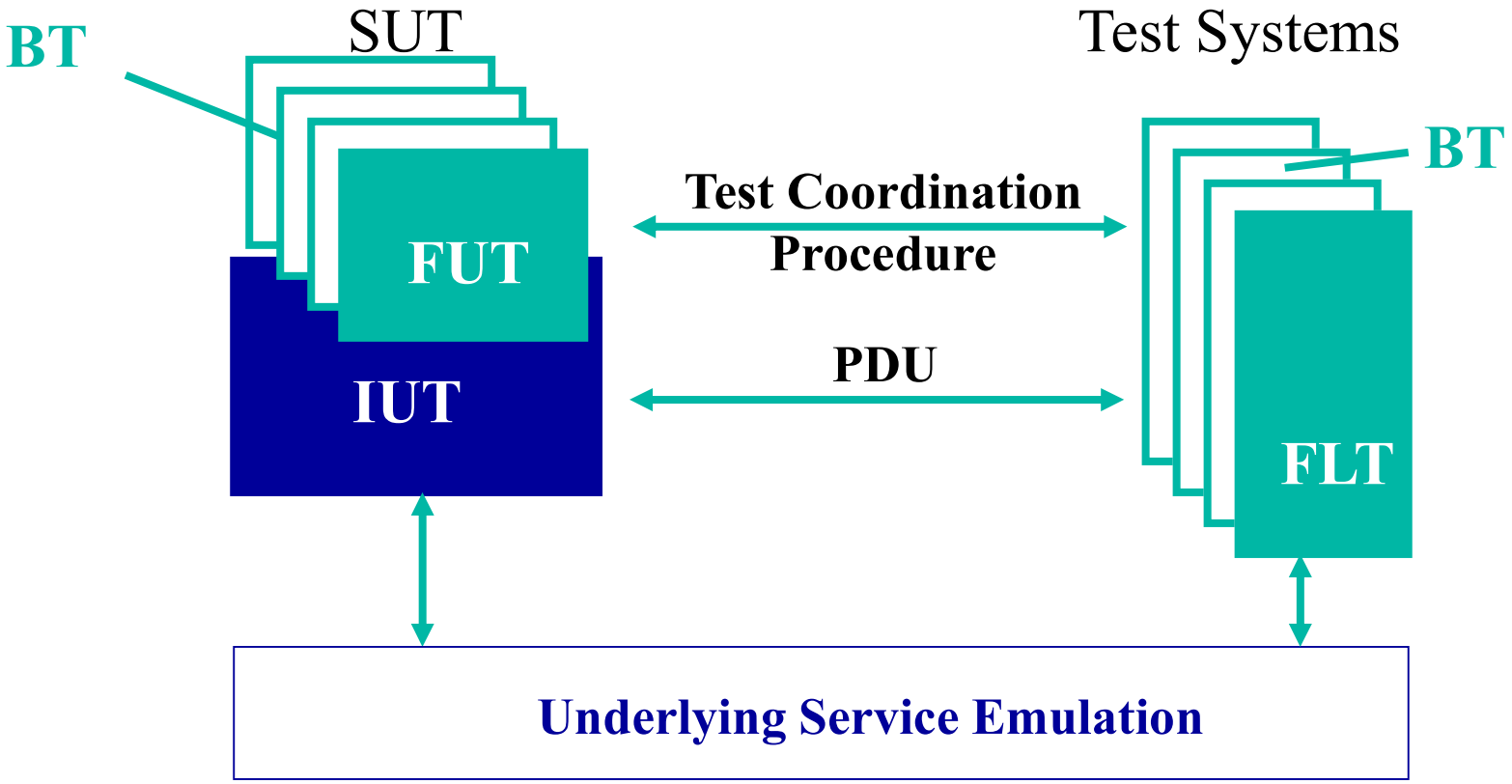


FT: Foreground Tester
BT: Background Tester
MP: Measurement Point

Performance Testing

- Performance Test Scenarios
  - Configuration of IUT, Foreground and Background Traffic
- Traffic Models
  - Description of discrete or continuous load patterns
- Measurement Points
  - Remote and Local Monitoring of Test Events and their Time Stamps
- Measurements and Analysis
  - Measuring the Delay and Rates of Test Events
- Performance Constraints and Performance Verdicts
  - Assessment of Performance Requirements and the Measured Performance

Protocol Performance Test Configuration



So what is different in wireless ?

Services in Wireless Environments ...

- Not only services known from the wired world
  - For applications from the wired world one has to prove that a service is valid also in a wireless environment
  - new services are expected for picoradio networks - carefull service definition needed!
- Service testing
  - either directly with a full-fledged emulator for the wireless environment
  - or via protocol testing of the service-relevant behavior subset of a protocol implementation

Wireless Communication Protocols ...

- Are just communication protocols - supporting services
- Operate in a non-deterministic environments
- Operate in a dynamic environments
- shall be self-stabilizing - this is more difficult to achieve
- but: are there any differences in testing??

Variability

- Given by high transmission loss rate
- Impact on testing
  - Fail does not mean incorrect behavior
- Testing
  - Emulate environment with bursty loss pattern in order to get a „controllable“ unreliability to back the test results
  - Use statistical testing to address the probabilities during test execution
    - a certain distribution of test results gives an overall test result

Variability (2nd)

- Given by discontinuous service
- Impact on testing
  - Known status of service availability as a precondition for test execution
- Testing
  - Emulate environment with available and inavailable services
  - Preamble can be used for this issue

Dynamic Environment Structure

- Given by reconfiguration and ad-hoc routing
- Impact on testing
  - Test components have to be created, removed and connected dynamically
- Testing
  - Emulation of dynamic environments toenforce change of routes, forwarding, handover
  - Access to configuration information of the IUT/environment

Energy-Dependent Behavior

- Impact on testing
  - Battery thresholds /operation mode might have impact on the expected behavior and their assessment (e.g. sleeping, e.g. selection of power level/packet length)
- Testing
  - Ideal would be to get a direct battery control to emulate efficiently the battery state /power saving modes
  - At least, observe the state of the IUT battery/saving mode to direct the test execution
  - How to achieve/enforce synchronization?

Testing the Energy Usage?

- What about checking the energy usage claims? (how many miles out of the galone?)
- Some “work cycle” has to be defined...and executed.
- In addition to functional and performance requirements...
- See you again in the requirement section...

Monitoring Sensitivity

- Wireless: observations are location dependent (!)
- Impact on testing
  - Monitoring parts of the test system **must** be located directly within the SUT (i.e. the mobile) - forget passive observation of a medium by third party!!
- Therefore:
  - Interferences resulting from the strong coupling of the test system with the tested system have to be analysed in order to prevent misinterpretation of test results

Specification-Based Testing ...

- Allows to reason about
  - Soundness and
  - Completeness of tests

- Soundness: what is a correct implementation for a given specification
- Completeness is not practical, coverage criteria are used instead

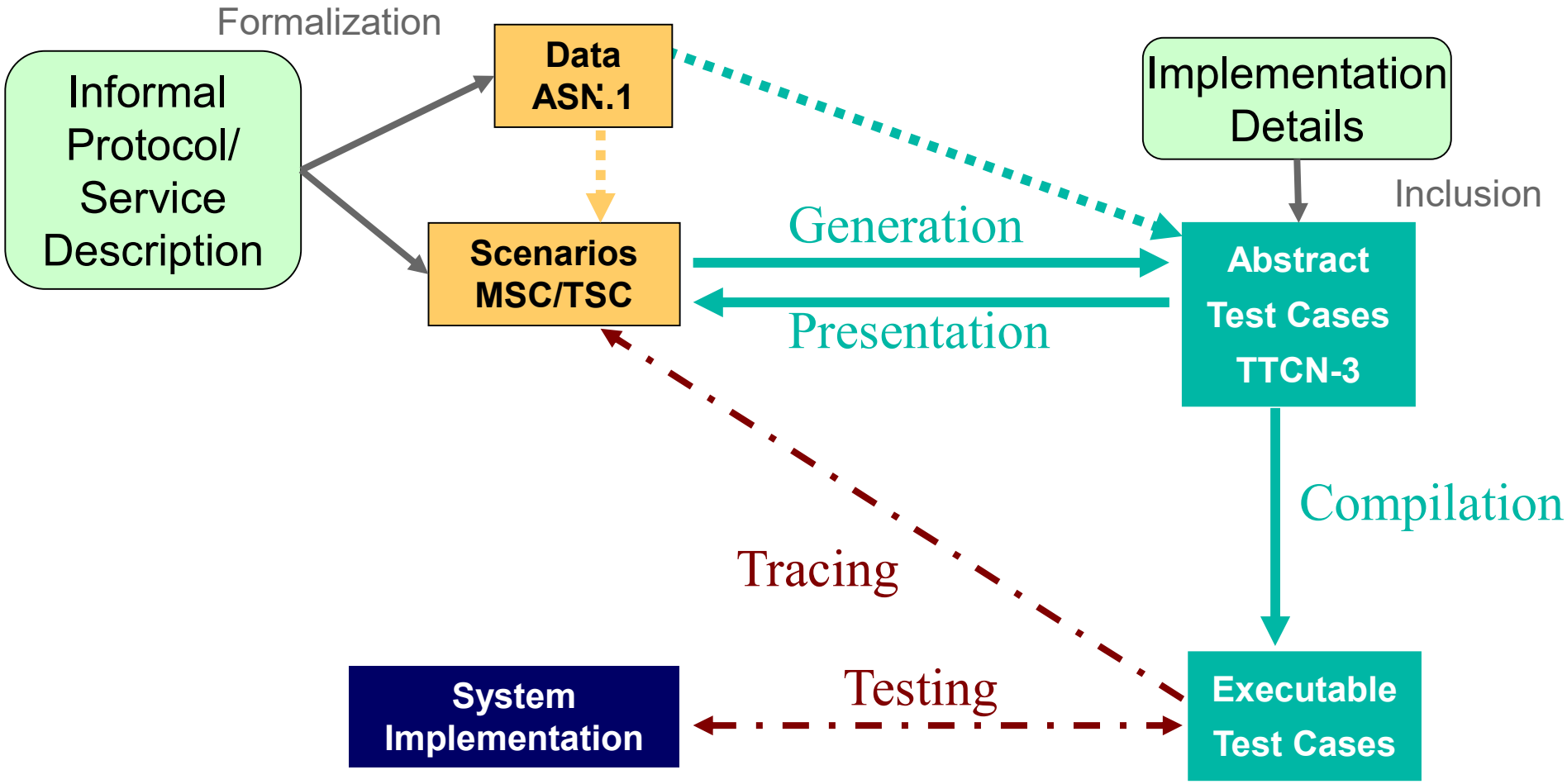
Coverage Considerations

- Is based on specification and fault models
- Gives indicators on the completeness of tests
- Real behavior specifications of are large
- Complete coverage can be realized only for scenarios of required/forbidden behavior not for complete behavior specification

Scenario Specification

- Message passing between interacting instances
- Makes use of the data specification
- Means to describe timing requirements and timed measurements

Specification-Based Testing



Message Sequence Charts (MSC)

- MSC is a graphical, formal scenario language that
  - Describes interaction between message-passing instances.
  - Supports **incomplete** (!) and complete specifications.
  - Supports structured design.
- Is used for
  - requirements specifications,
  - abstract, first shot service descriptions,
  - test purposes in conformance testing, ...

MSC-2000

- MSC-2000 (ITU-T in Z.120) adds:
  - Object-orientation
  - Use of data in the language of user's choice
  - Improved time and structural concepts
  - Method calls
- No notion of energy consumption
- No notion of service continuity...

Time in MSC-2000

- Supports the notion of quantified time for the description of real-time systems
- Precise meaning of the sequence of events in time
- MSC events are instantaneous
- Time constraints can be specified in order to define the time at which events may occur
- Time measurements can be specified in order to observe the time at which events occured

Tree and Tabular Combined Notation (TTCN)

- TTCN is a testing language
  - describes interactions between testers and implementations under test in terms of message exchange at PCOs (Points of Control and Observation).
  - is part of the Conformance Testing Methodology and Framework on testing of OSI systems.
  - Concurrent TTCN addresses testing with parallel test components.
- Defined by ISO in ISO 9646-3 and by ITU-T in X.292 (known as TTCN-2)

PerfTTCN

- Notation that
  - describes performance tests understandable and re-usable, makes the test results comparable
  - is an extension to conformance test descriptions with notions of time and performance
- PerfTTCN
  - based on TTCN, 2nd Edition with
    - Background and Foreground Test Components
    - Traffic Models
    - Measurements and Analysis
    - Performance Constraints and Performance Verdicts

TTCN 3rd Edition

- Major **changes** to TTCN-2:
  - Concentration on the essence of a testing language
  - Definition of a programming-like core language
  - Improved modularization
  - Improved integration of external data specifications

- Major **extensions** to TTCN-2:
  - Dynamic test configurations
  - Enhanced communication mechanism
  - Control for test execution

TTCN 3rd Edition (2)

- (Recognized) Open Issues
  - real-time test concepts
  - performance test concepts
  - statistical test concepts

  - could be incorporated via external functions (needs however further elaboration)

  - What about energy??

Test Sequence Charts

- A graphical representation format for TTCN-3
- Supports:
  - high-level sequence charts with behavioral operators
  - hyper sequence charts for improved readability
  - hybrid sequence charts for the inclusion of native TTCN

Conclusions

- Testing wireless communication protocols
  - can reuse existing test concepts
  - but impose new requirements for test setup
    - variability, reconfigurability, ability to monitor
  - and for test result analysis
    - stochastic nature of test results
- Test tools based on MSC/TSC and TTCN solve some but not all issues
- Introduce energy in specification/requirements!