

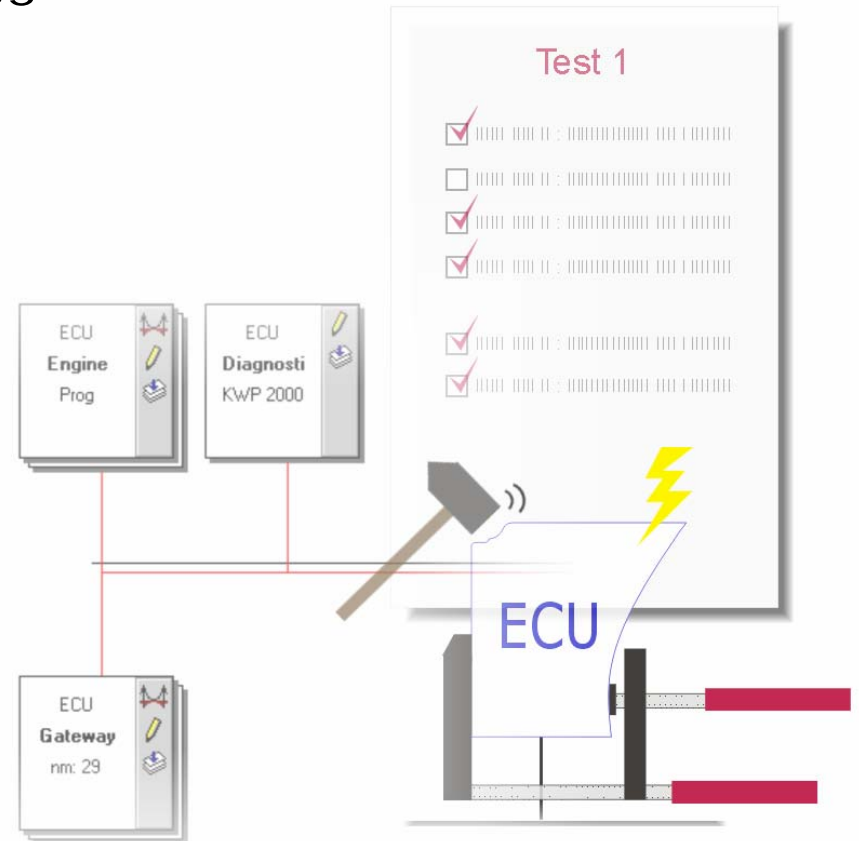


## Conformance Testing of LIN ECUs

Vector LIN Symposium 2008

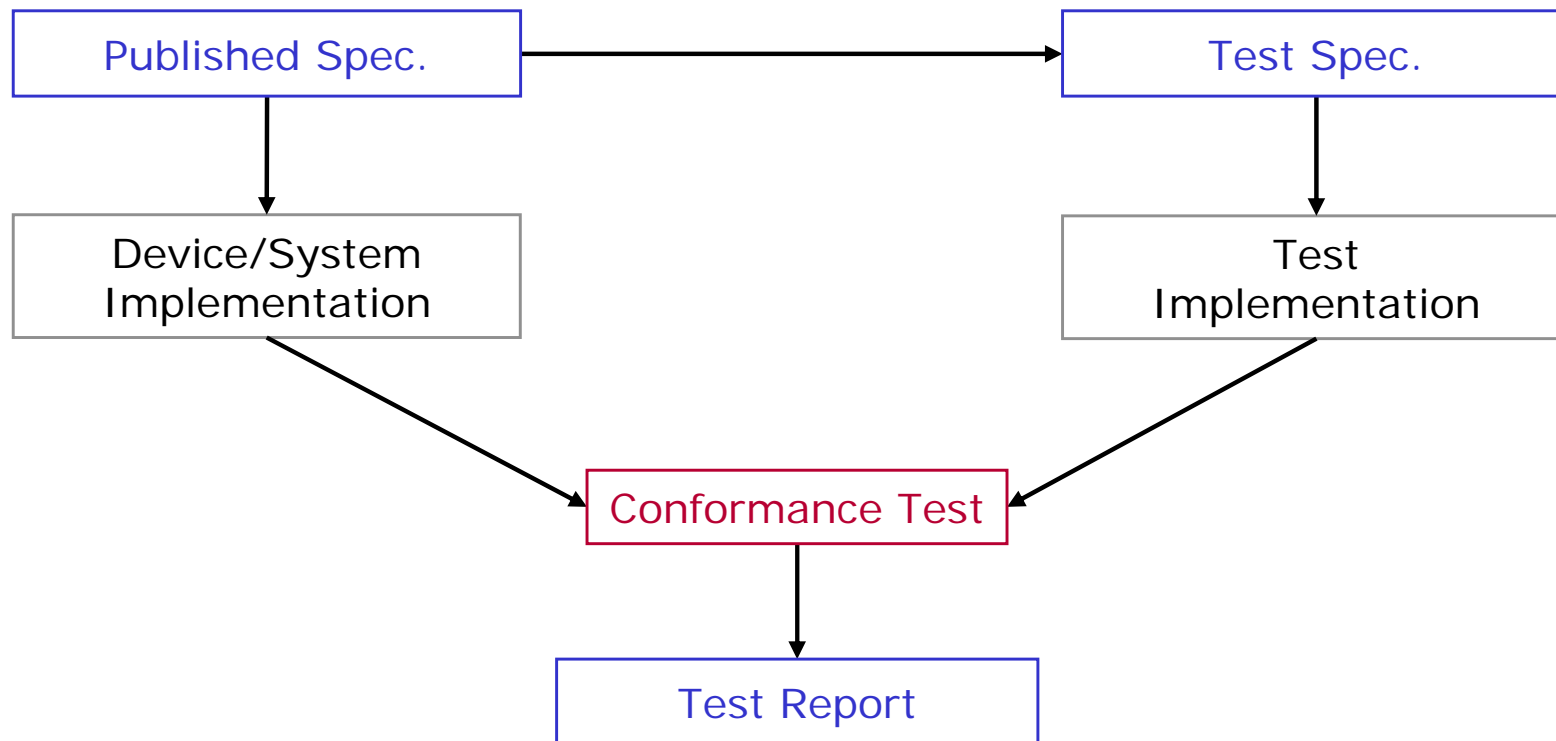
Gavin C. Rogers, Vector Informatik GmbH

- ❑ Introduction
- ❑ Test Strategies and Methods for ECUs
- ❑ LIN Conformance Tests
- ❑ Slave Conformance Test
- ❑ Master Conformance Test
- ❑ Summary & Outlook

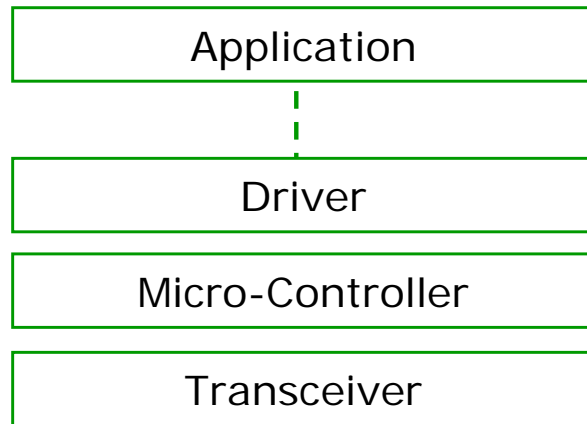


### ❑ What is a Conformance Test?

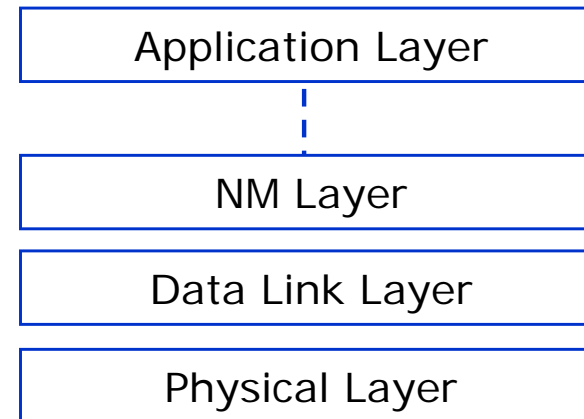
- ❑ Test of device or system compliance according to a standard or published specification
- ❑ Basis for certification or accreditation by a test institution



- ❑ What test strategies exist for ECUs?
  - ❑ Component Testing
    - ❑ Dependent on specific HW implementation and interfaces
  - ❑ OSI-Layer Testing
    - ❑ Independent of HW implementation and/or interfaces



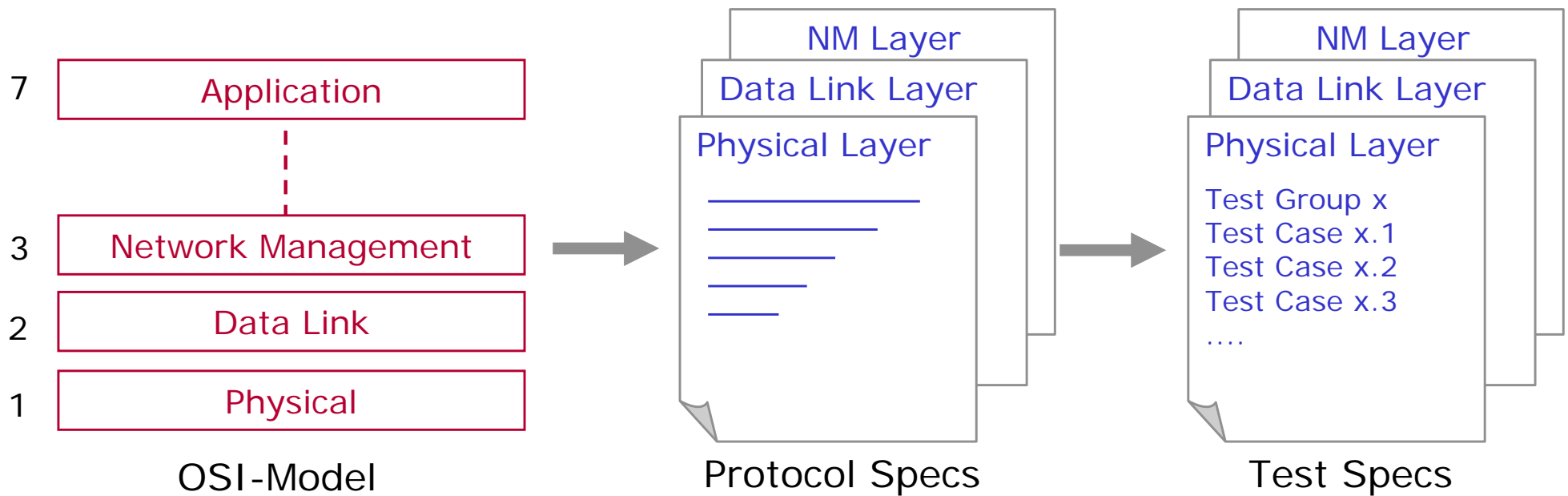
Component-Model of ECU



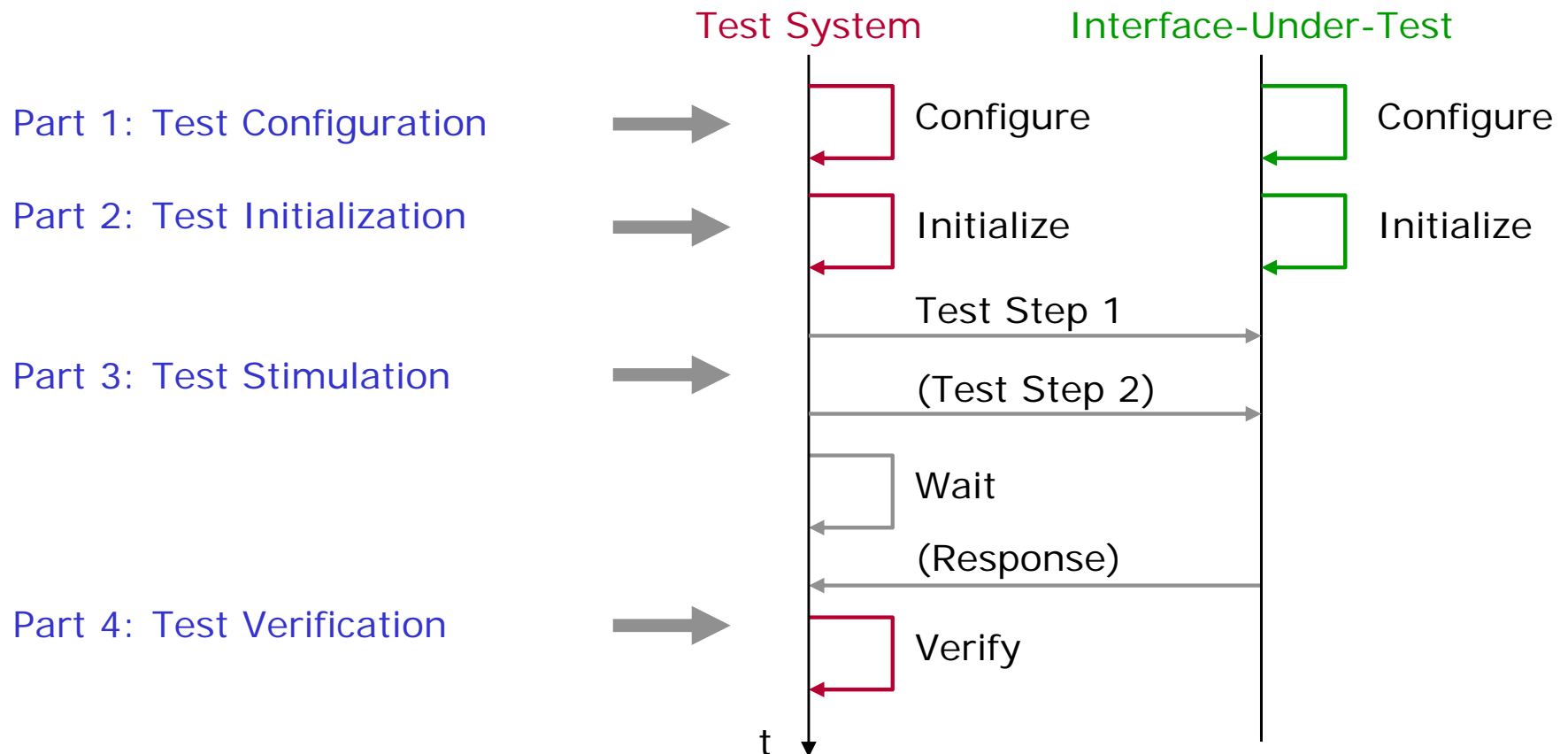
OSI-Model of ECU

- ❑ What test methods exist for ECUs?
  - ❑ Black Box Test
    - ❑ Stimulation and verification via ECU's external interfaces only
      - ❑ e.g. LIN or CAN interface of ECU
  - ❑ White Box Test
    - ❑ Stimulation and verification via ECU's internal interfaces only
      - ❑ e.g. LIN driver API (standardized)
  - ❑ Gray Box Test
    - ❑ Stimulation and verification via ECU's external and internal interfaces
      - ❑ e.g. LIN interface of ECU and LIN driver API

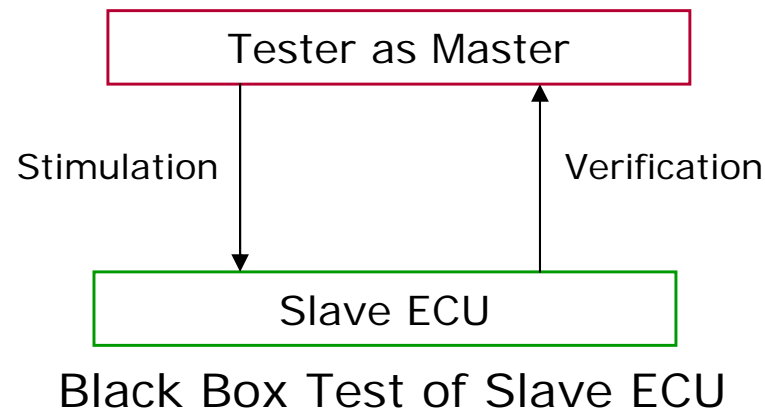
- ❑ How are LIN conformance tests specified?
  - ❑ Specified by LIN consortium for OSI layers 1, 2 and 3
  - ❑ Available for LIN1.3, LIN2.0 and expected for LIN2.1 Q2 2008
  - ❑ Defined as a series of test groups and test cases
  - ❑ Different test cases for Master and Slave components/nodes



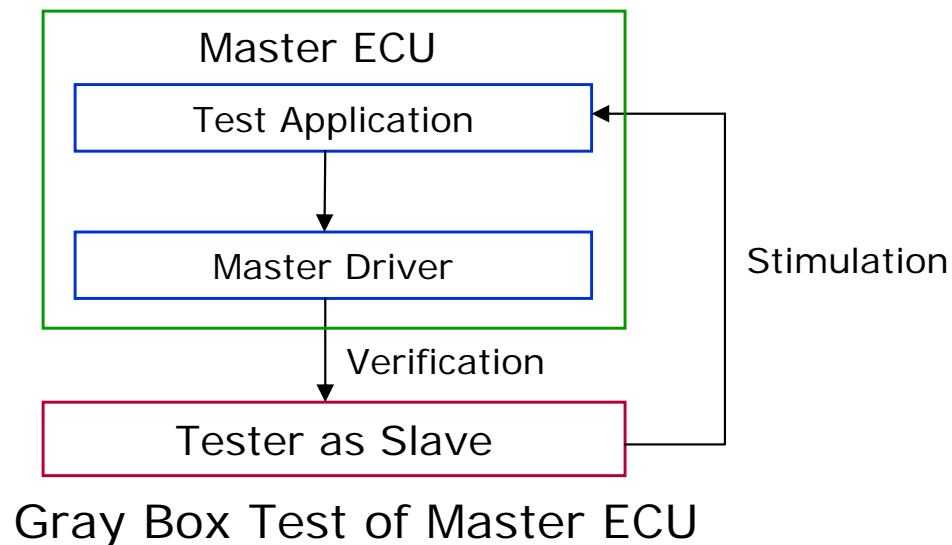
- ❑ How are LIN conformance test cases defined?
  - ❑ Each test case consisting of 4 parts
  - ❑ Test stimulation and/or verification part not always defined



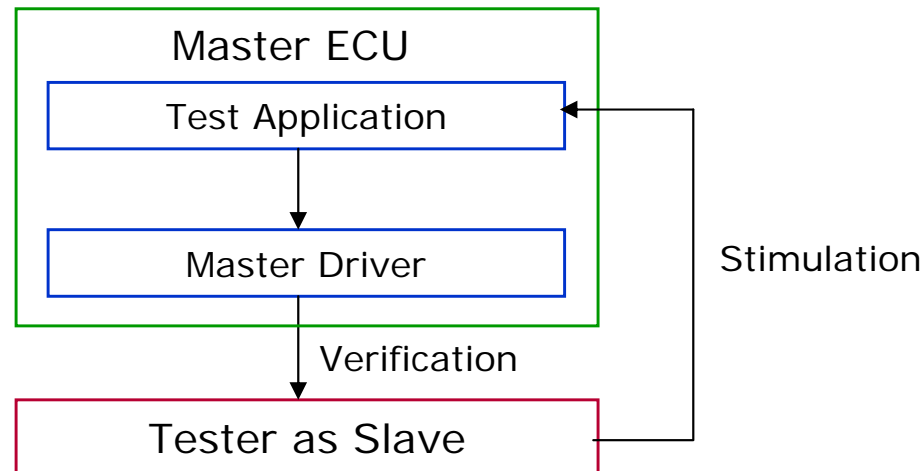
- ❑ How are Slave tests implemented in CANoe.LIN?
  - ❑ As black box tests
  - ❑ Stimulation/verification mostly possible via LIN interface of ECU
  - ❑ Some tests require application-specific stimulation/verification e.g.
    - ❑ Stimulation of wakeup frame(s)
    - ❑ Stimulation of event-triggered frame
    - ❑ Verification of sleep command and bus idle detection



- ❑ How are Master tests implemented in CANoe.LIN?
  - ❑ As gray box tests
  - ❑ Configuration of Master driver according to special test LDF
  - ❑ Indirect stimulation of LIN driver via special test application
    - ❑ Provided test application code uses standardized LIN driver API
  - ❑ Verification via LIN interface of ECU



- ❑ What disadvantages does a gray box Master test have?
  - ❑ Driver configuration for test and development is not identical
    - ❑ i.e. Master cannot be tested during all phases of development
  - ❑ Considerable test preparation effort required e.g.
    - ❑ Configuration of driver according to special test LDF
    - ❑ Compilation and download of driver and test application code to HW
  - ❑ OEM cannot perform test without support from supplier
    - ❑ Compiler and download tools required for Master HW



Gray Box Test of Master ECU

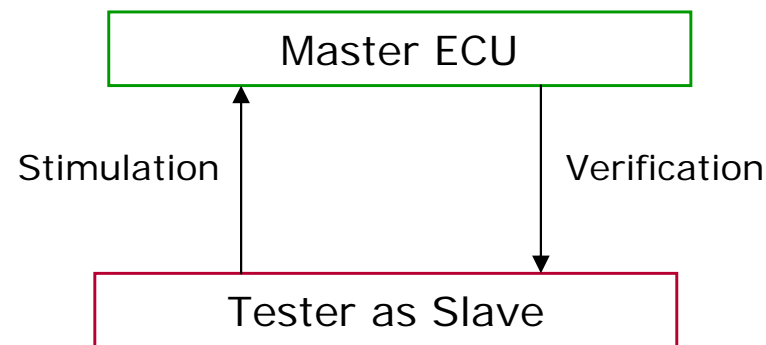
### ❑ Is a black box Master test realizable?

#### ❑ Motivation

- ❑ Applicable during all phases of Master ECU development
- ❑ Executable by OEMs and suppliers with minimal preparation time

#### ❑ Requirements

- ❑ Identical driver software for test and development
- ❑ Direct stimulation and verification of Master driver by tester (Slave)
  - ❑ i.e. without Master test application
- ❑ Stimulation and verification of all tests via LIN



Black Box Test of Master ECU

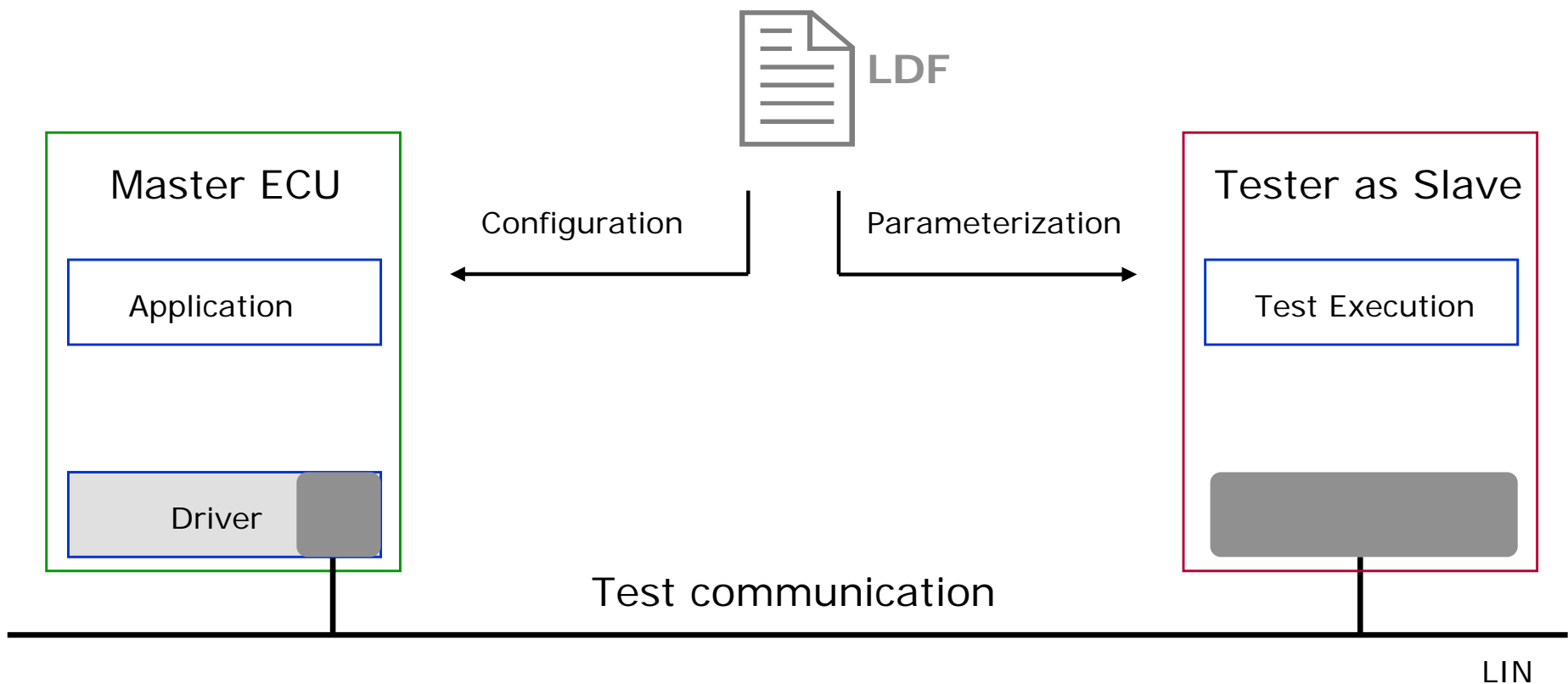
# Master Conformance Test

## Black Box Test

### ❑ Is a black box Master test realizable?

#### ❑ Proposed solution

Stimulation and verification of Master ECU via driver extension with test interface

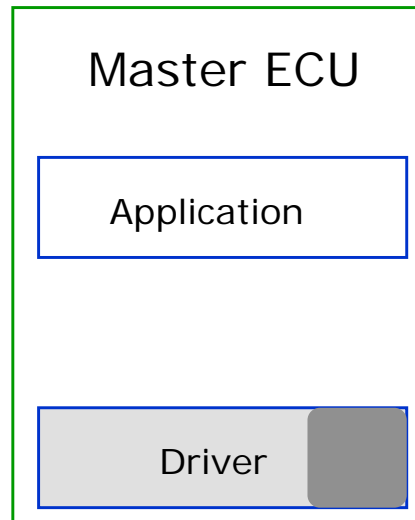


- ❑ What type of stimulation/verification is required via test interface?

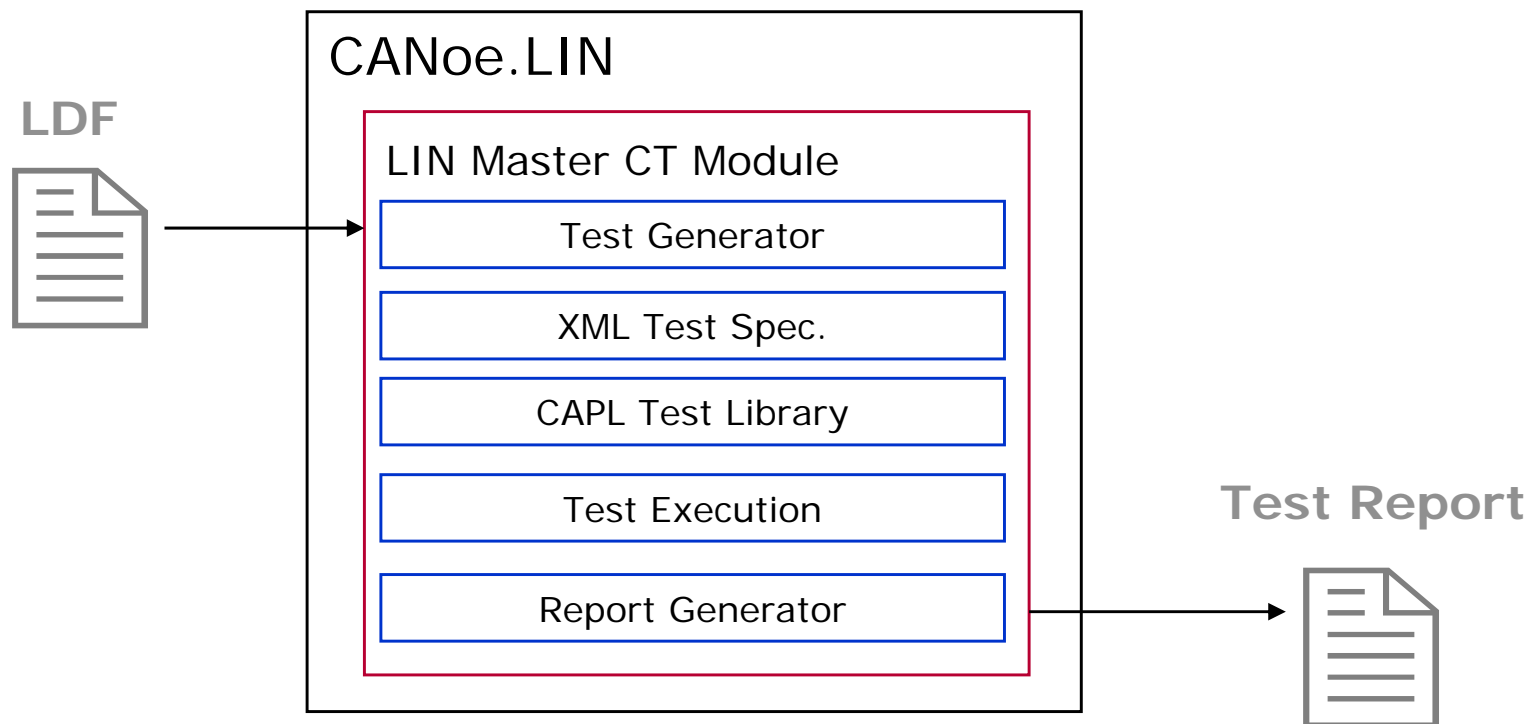
Test Case (LIN 2.0)	Stimulation	Verification
DLL 2.1, 2.3, 2.5, 2.7	Send reconf. command	
DLL 2.10.2, 3.1.1, 3.1.3, 3.2.2	Set schedule table	
DLL 3.4	Send reconf. command	
DLL 3.6	Send reconf. command	Read driver status word
NCNM 3.1, 3.3	Send sleep command	Read driver status word
NCNM 3.5.1	Send wakeup	Read driver status word

- ❑ Following test services must be supported by Master driver extension
  - ❑ Send reconfiguration command
  - ❑ Set schedule table
  - ❑ Send sleep command
  - ❑ Send wakeup
  - ❑ Read driver status word

- ❑ What other requirements must the Master driver extension meet?
  - ❑ Test communication via LIN to stimulate Master from tester (Slave)
    - ❑ New diagnostic service e.g. SID 0xB8
    - ❑ TP single frame communication i.e. MasterReq and SlaveResp
  - ❑ Minimal resources
  - ❑ Code for test support must be removable before production
    - ❑ Via preprocessor define



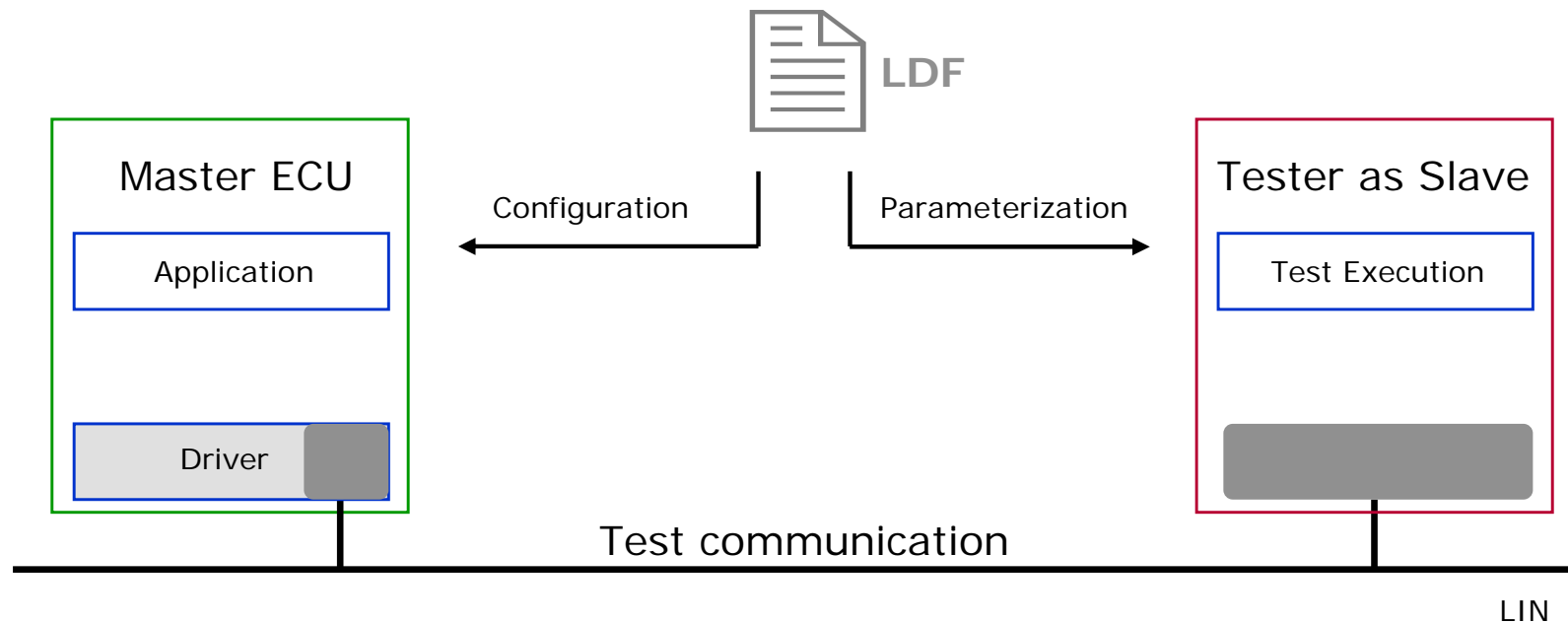
- ❑ What requirements must CANoe.LIN as test system meet?
  - ❑ Configuration of relevant test cases
  - ❑ Parameterization of test cases
  - ❑ Test implementation supporting test communication
  - ❑ Report generation



# Conformance Testing of LIN ECUs

## Summary

- ❑ LIN Slave Conformance Test
  - ❑ Black box test available for LIN1.3, LIN2.0 and J2602 with CANoe.LIN 7.0
- ❑ LIN Master Conformance Test
  - ❑ Gray box test available for LIN2.0 with CANoe.LIN 7.0
  - ❑ Black box test would require Master driver with test interface



- ❑ When will CANoe.LIN support the LIN2.1 conformance tests?
  - ❑ Release of LIN2.1 CT specification expected in Q2 2008
  - ❑ Support of LIN2.1 Slave test planned for 7.0 SP4/5 (Q3 2008)
    - ❑ Preliminary support according to draft spec. already available with 7.0
  - ❑ Support for LIN2.1 Master test planned for 7.1 (Q4 2008)
    - ❑ To be implemented either as gray box and/or black box test
  
- ❑ Will CANoe also support OEM-specific tests?
  - ❑ OEM test packages for CAN and LIN planned as project license
    - ❑ Test package includes I/O hardware (if required)
    - ❑ Available only if requested by OEM and its suppliers

Thank you for your attention.

For detailed information about Vector  
and our products please have a look at:

[www.lin-solutions.com](http://www.lin-solutions.com)

**Author:**

Gavin C. Rogers, Peter Decker

Vector Informatik GmbH

Ingersheimer Str. 24

70499 Stuttgart