Frank Tränkle and Wolfgang Eismann / ETAS GmbH

LABCAR-AUTOMATION
Open Interfaces for Test Automation

TestingExpo 2005, Stuttgart

Outline

- Evolution of Test Automation
- Systematic Approach for Test Automation Framework
- Open Interfaces
- Summary

LABCAR-AUTOMATION

Motivation

Challenge at Customer Robert Bosch:

- More than 1000 automated system tests per year and per division
- Need to improve productivity
- Replace heterogeneous tool environment, standardize processes and tools

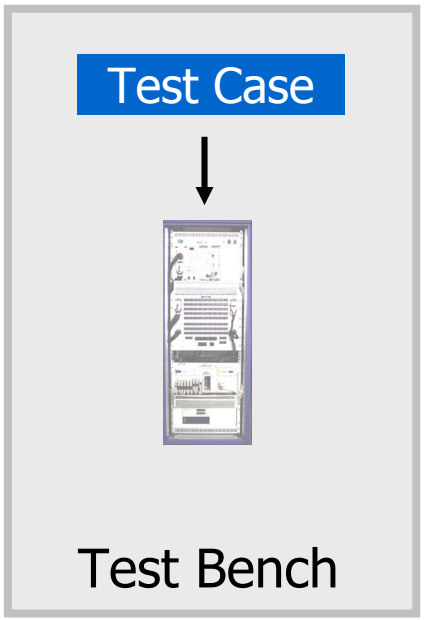
BOSCH

Solution: Provide new test automation tool, that …

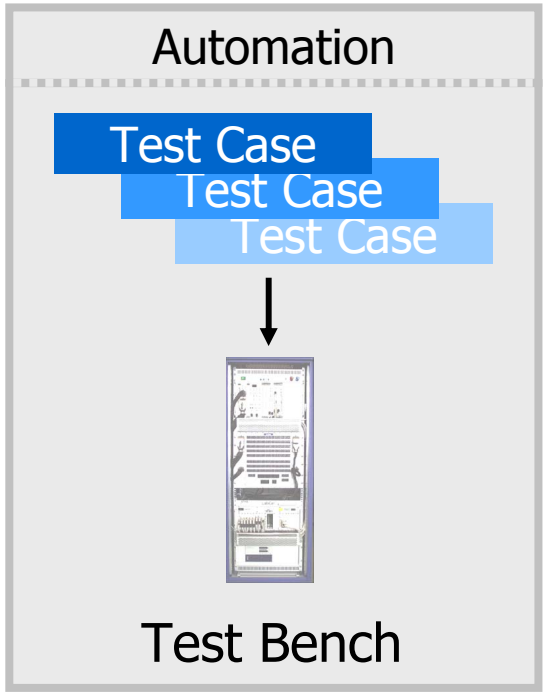
- Increases reusability of test cases
- Enables more transparent, manageable operation of test systems
- Offers high flexibility, so that it can be used as standardization platform for test automation

Evolution of Automated Testing

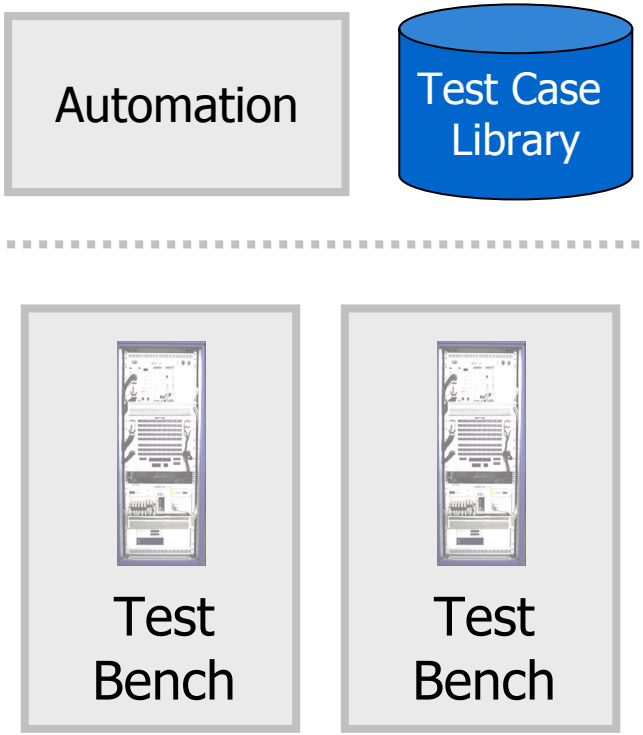
Interactive Testing



"Typical Automation"



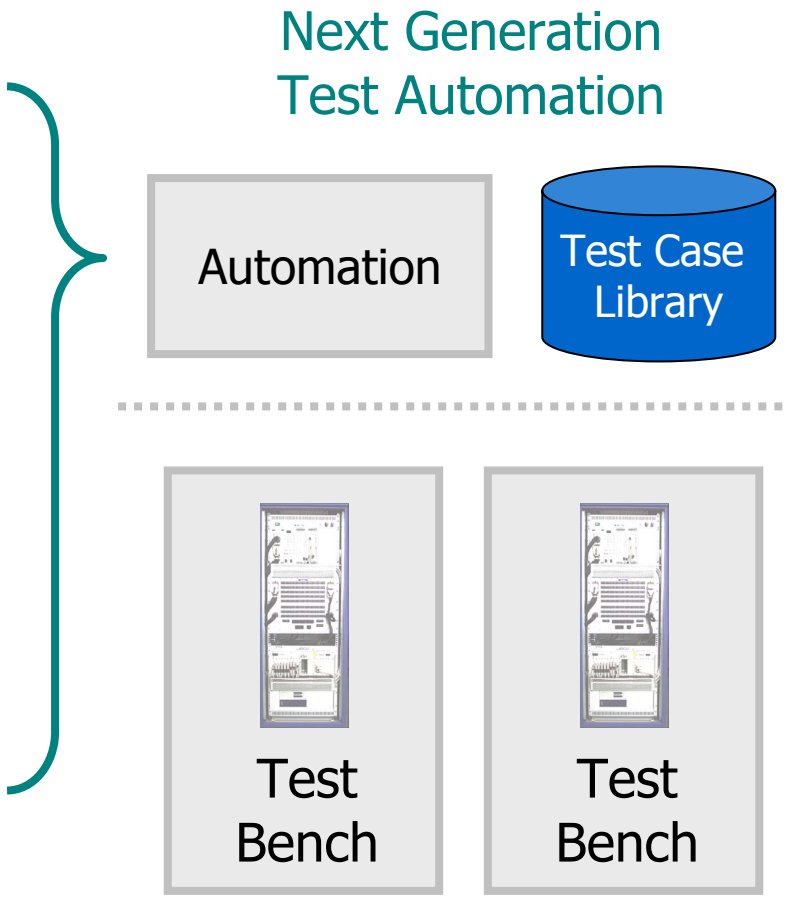
Next Generation Test Automation



Evolution of Automated Testing

LABCAR-AUTOMATION provides:

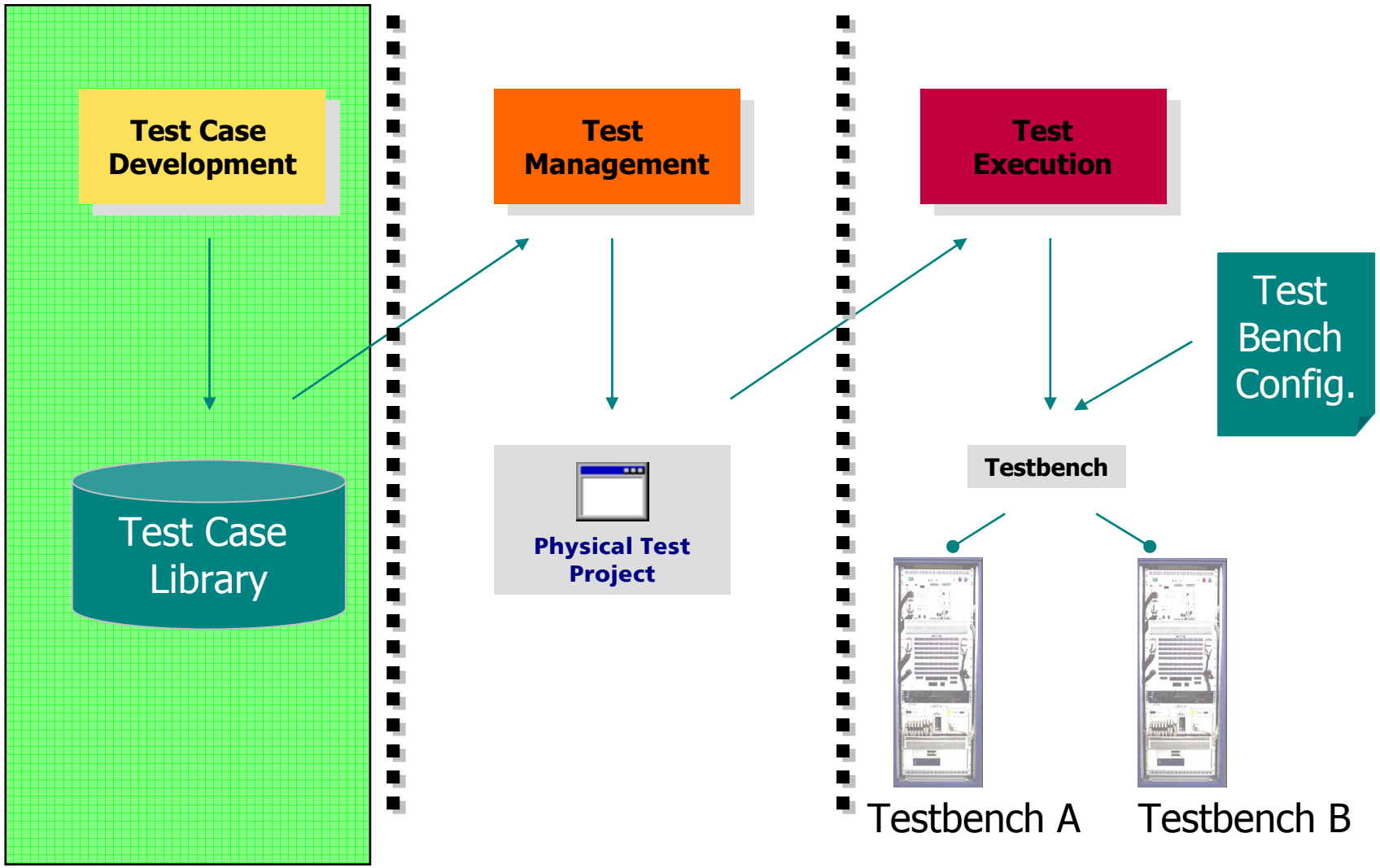
- Efficient development, deployment, and execution of reusable test cases
- Abstraction layer between test cases and test benches (test bench independence)
- 3rd party tool integration (open interfaces)
- Test design tool independence (matching various test case developer skills)



Outline

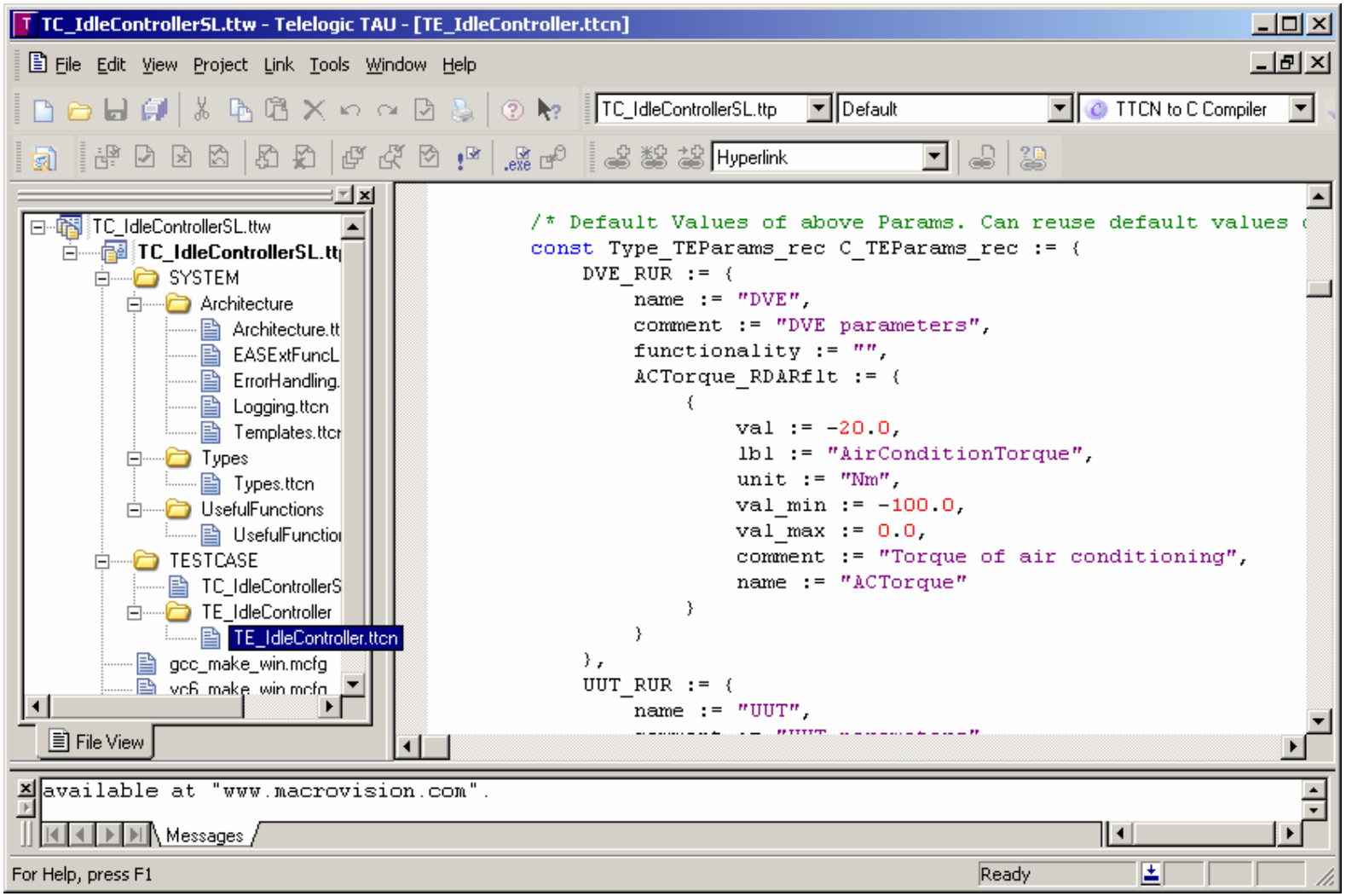
- Evolution of Test Automation
- Systematic Approach for Test Automation Framework
- Open Interfaces
- Summary

Test Case Development

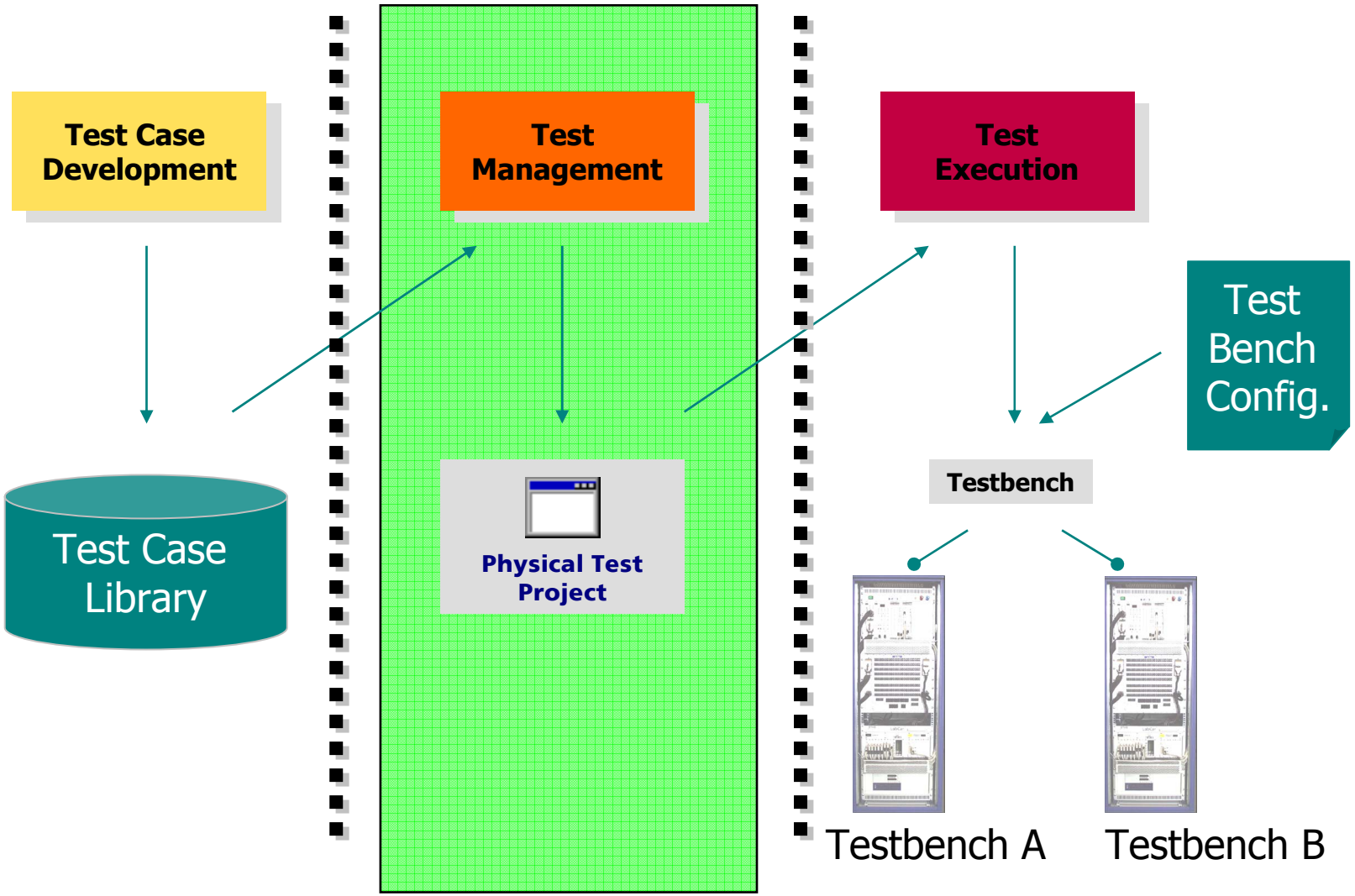


Test Case Development

Example: TTCN-3 Development with TAU/Tester

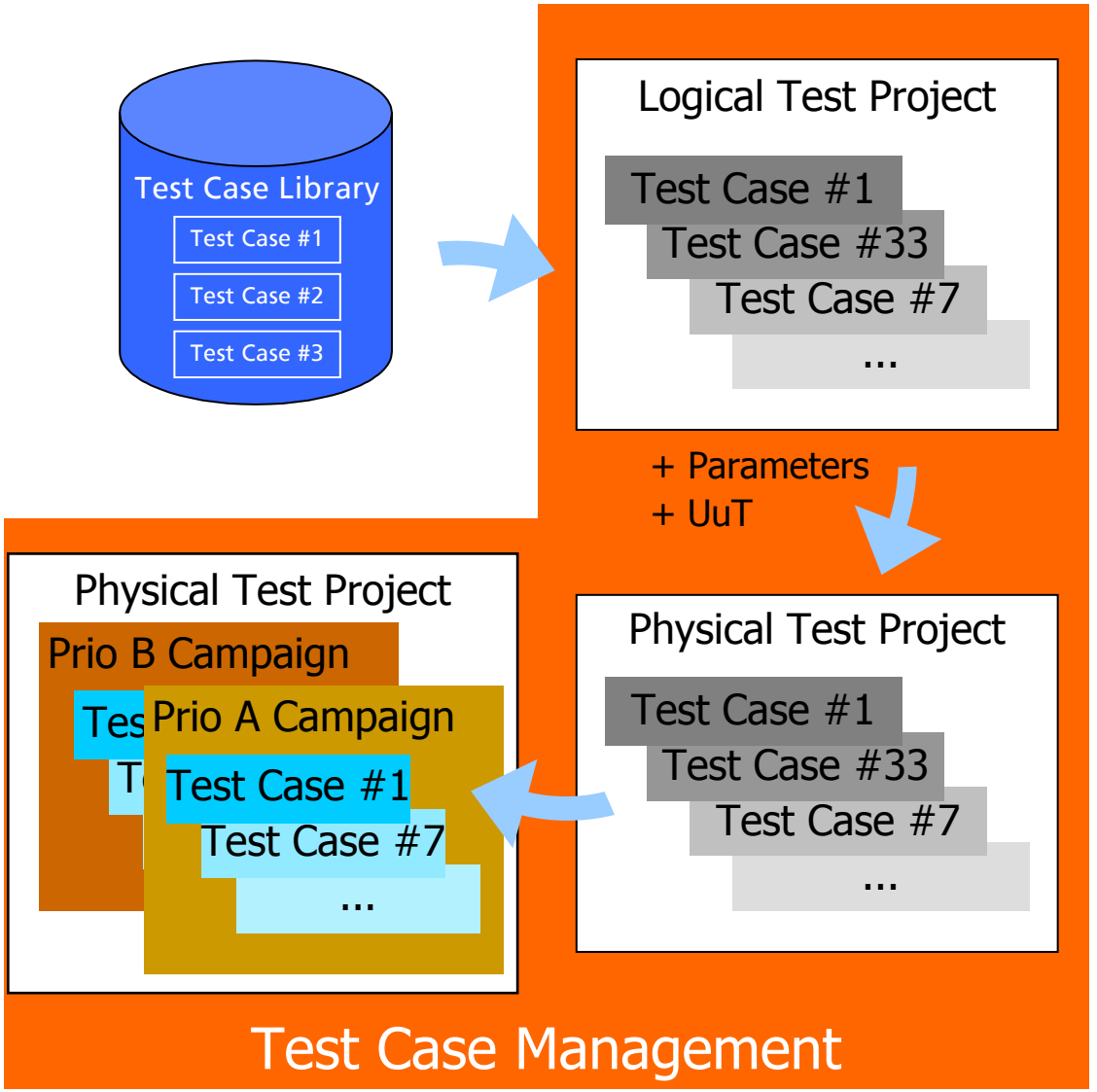


Test Management



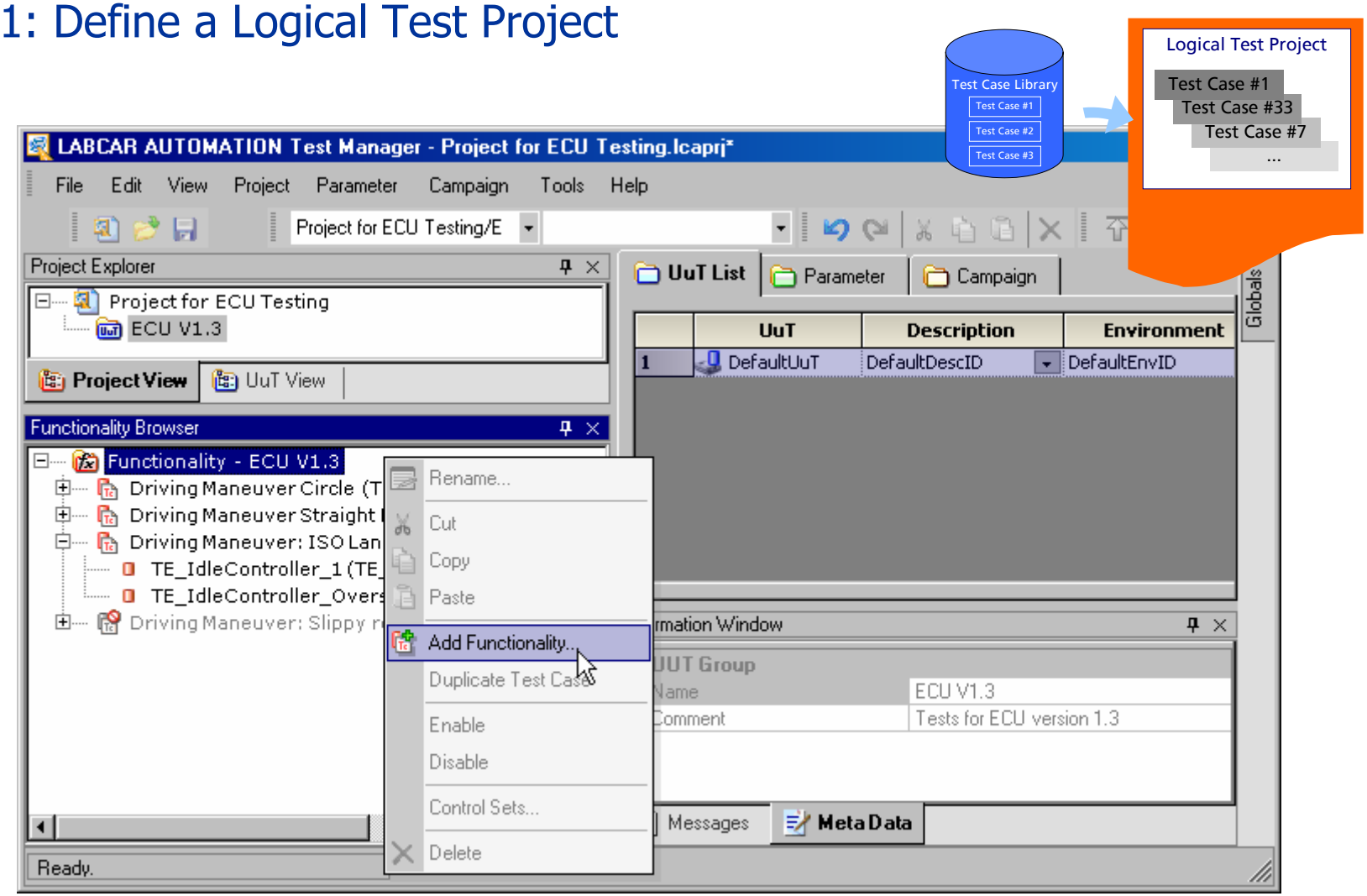
Test Management

- Set up logical test project by selecting test cases from library
- Convert logical test project into physical test project by assigning test parameters and ECU variants
- Define reproducible test runs for a test project
- Test bench configured at execution time of test case / test campaign



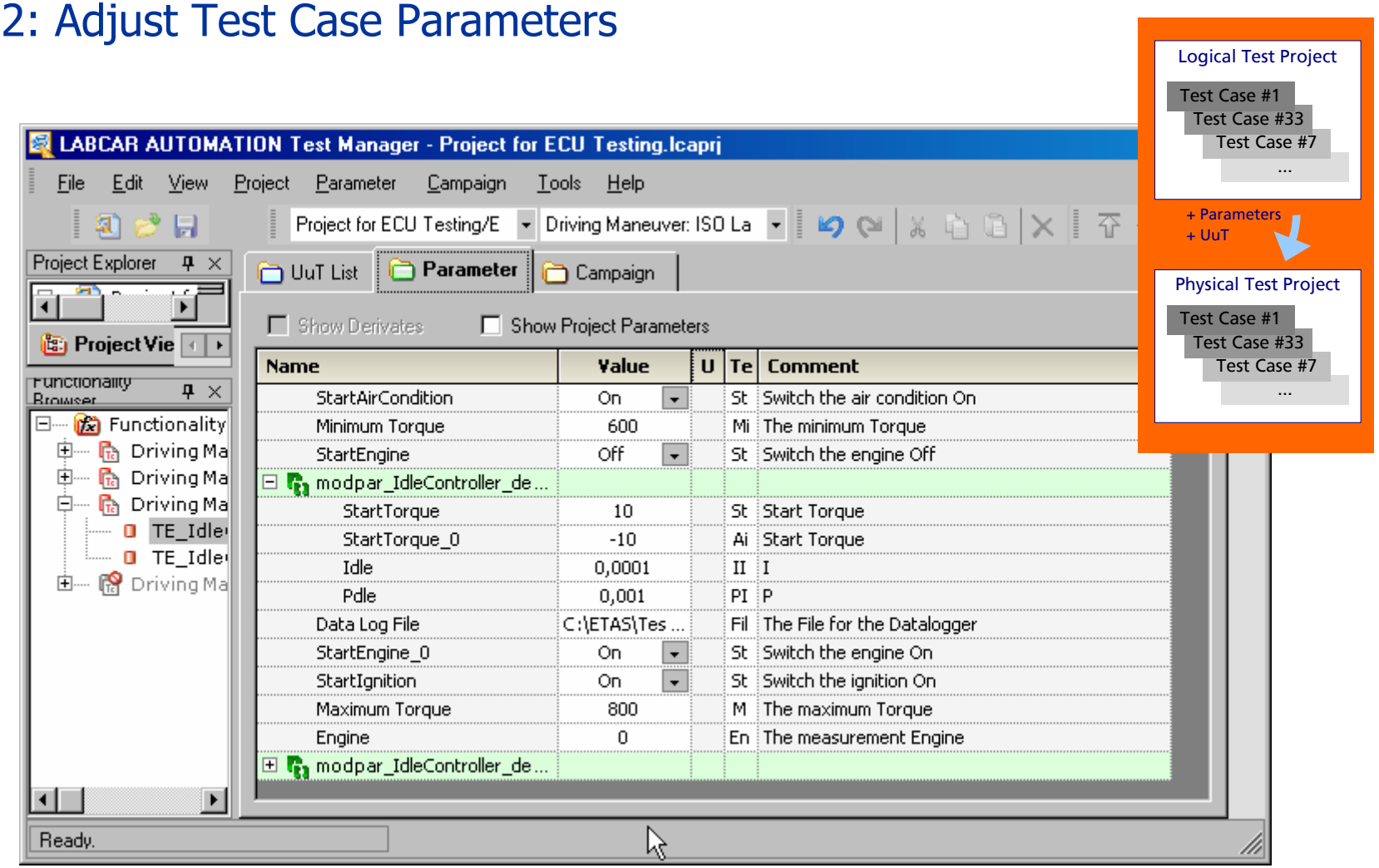
Test Management

Step 1: Define a Logical Test Project



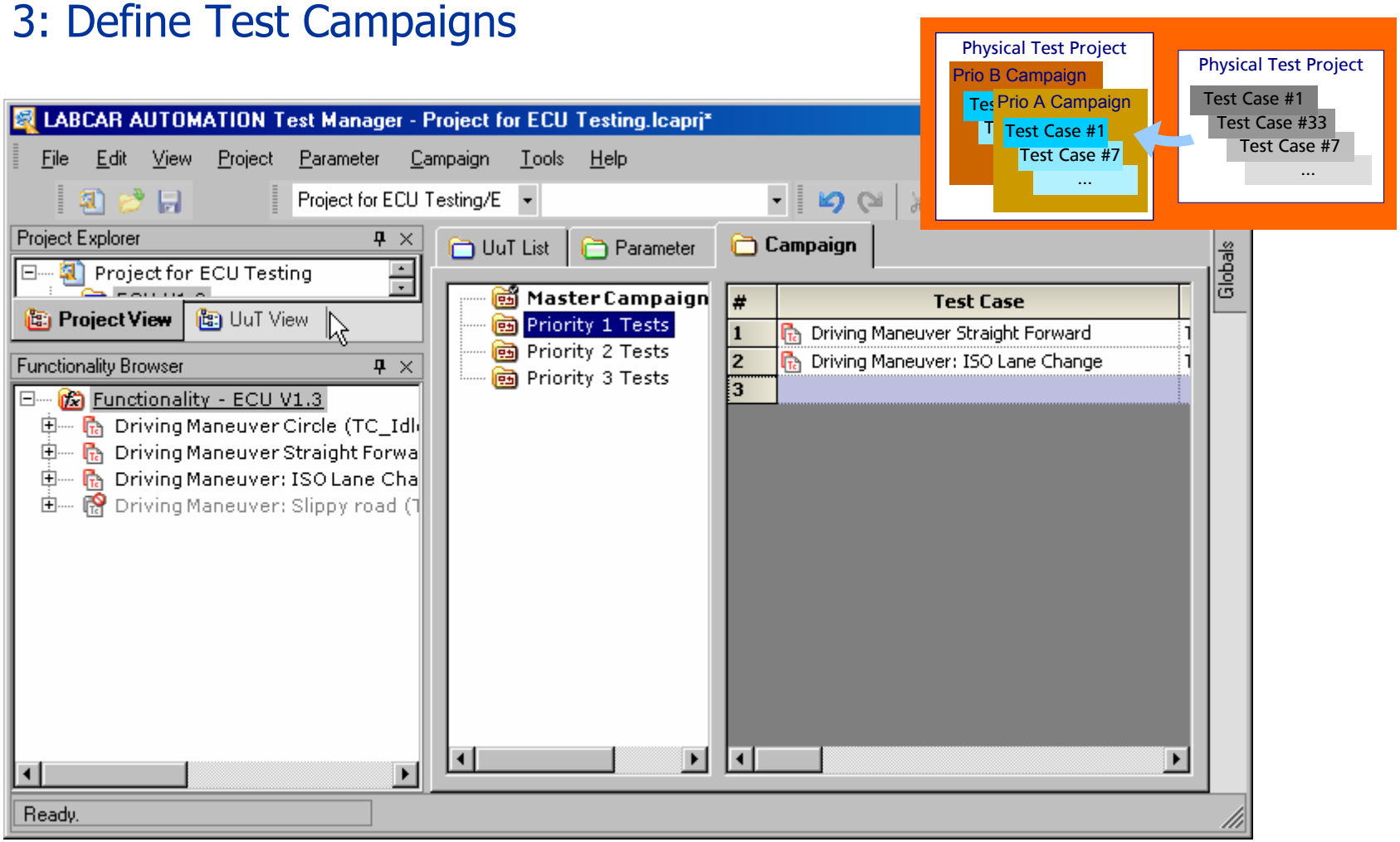
Test Management

Step 2: Adjust Test Case Parameters

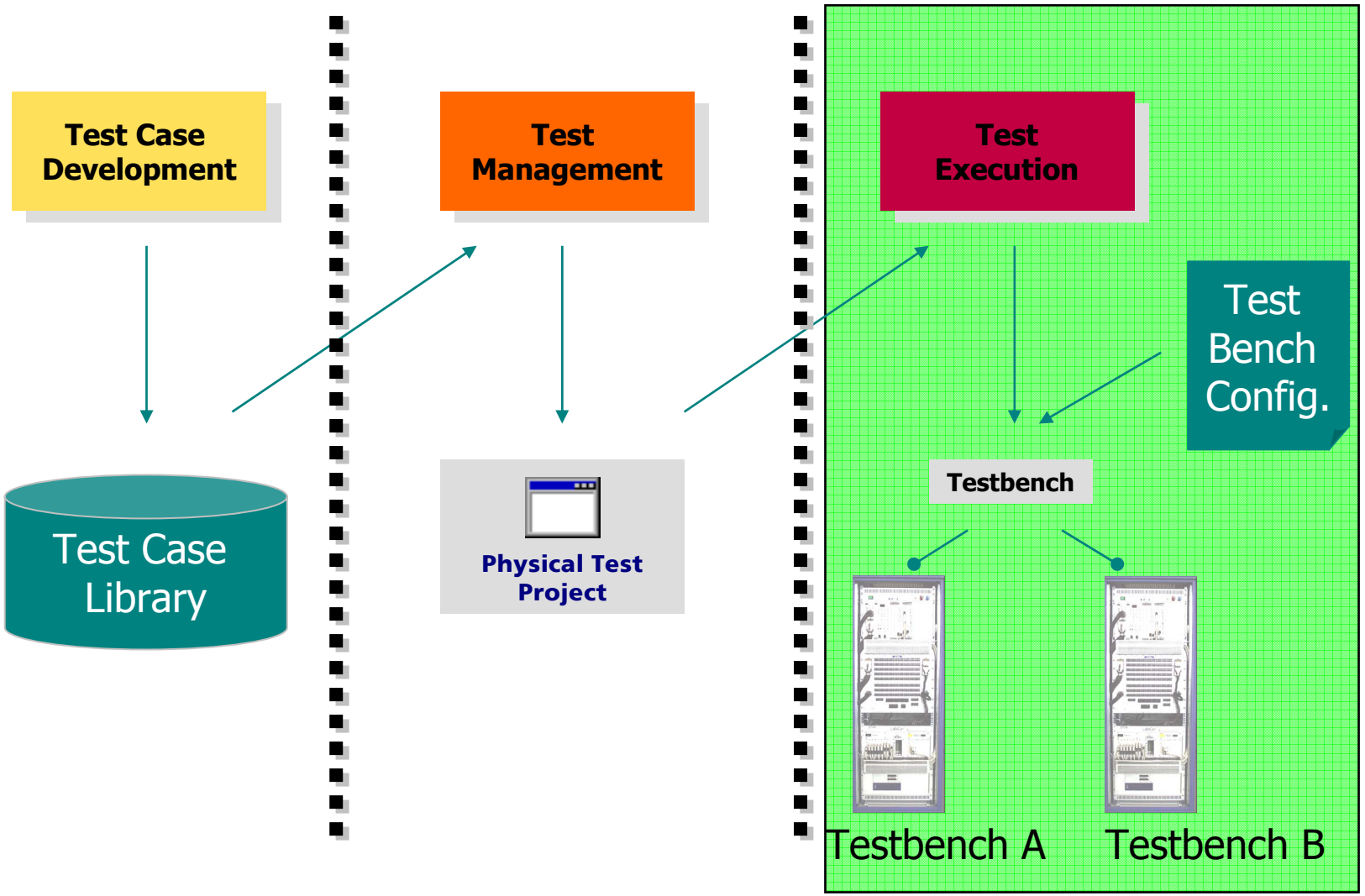


Test Management

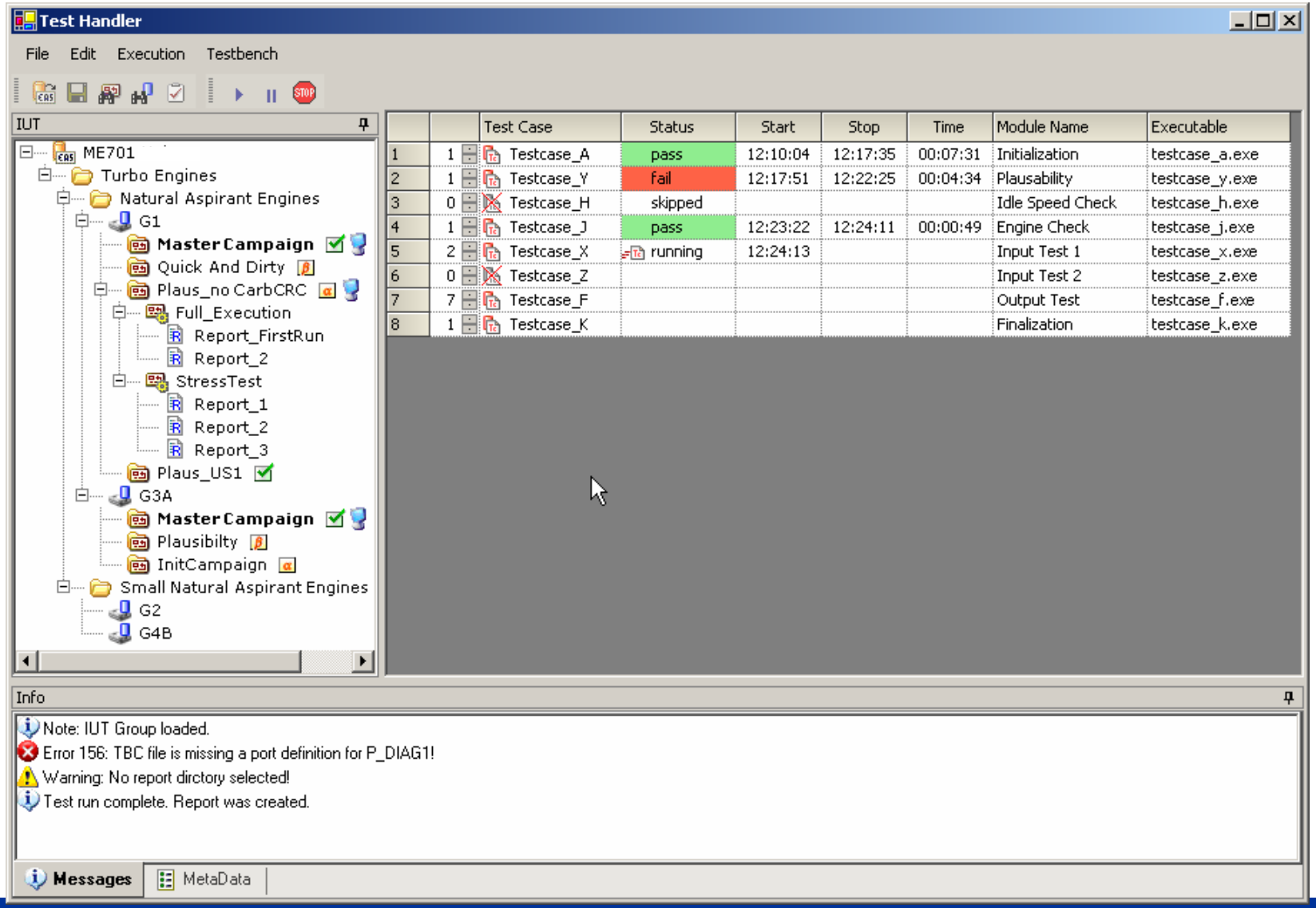
Step 3: Define Test Campaigns



Test Execution

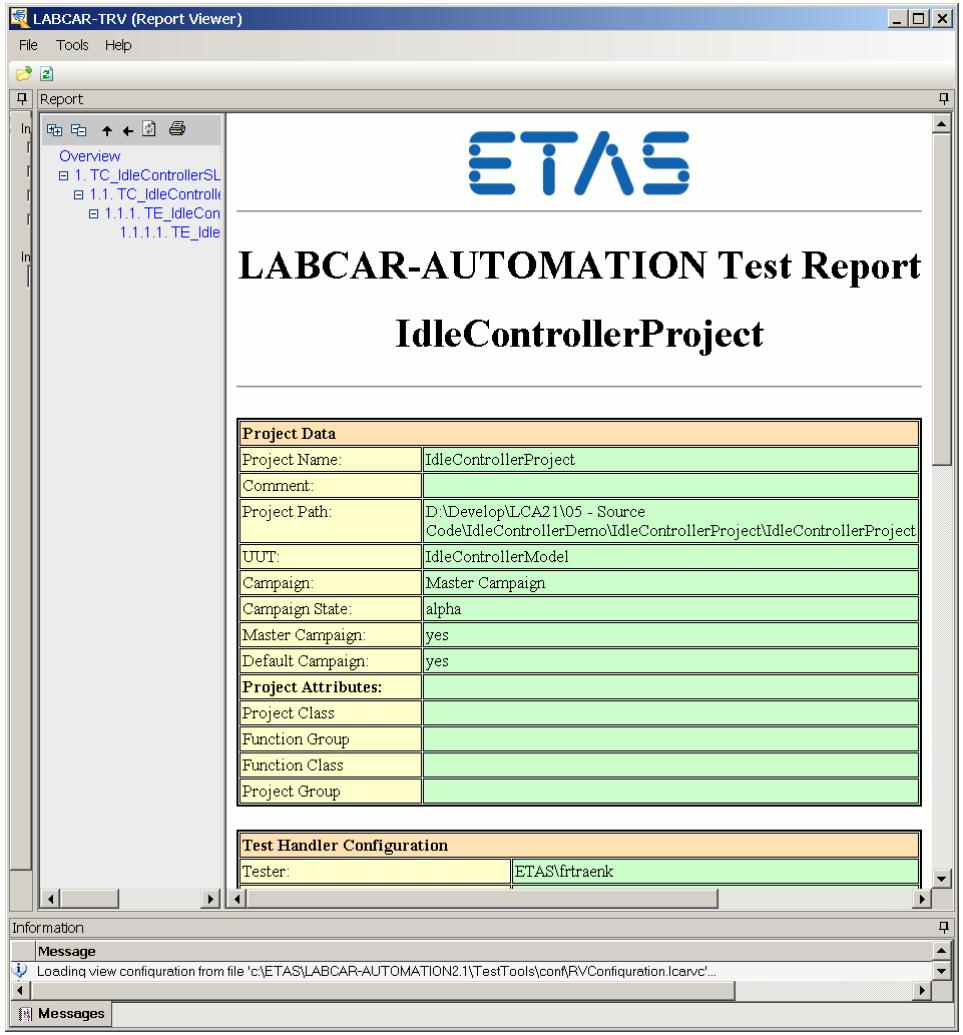


Test Execution



Test Execution

Hierarchical HTML View on XML Reports



Product Features Overview

Test Case Development Tools

- Open APIs for integration of 3rd party test case development tools and languages (e.g. Python)
- Integration package for Telelogic TAU/Tester development tool (TTCN-3 support)

Test Management Tools

- Project Management
- Parameter Management
- Campaign Management

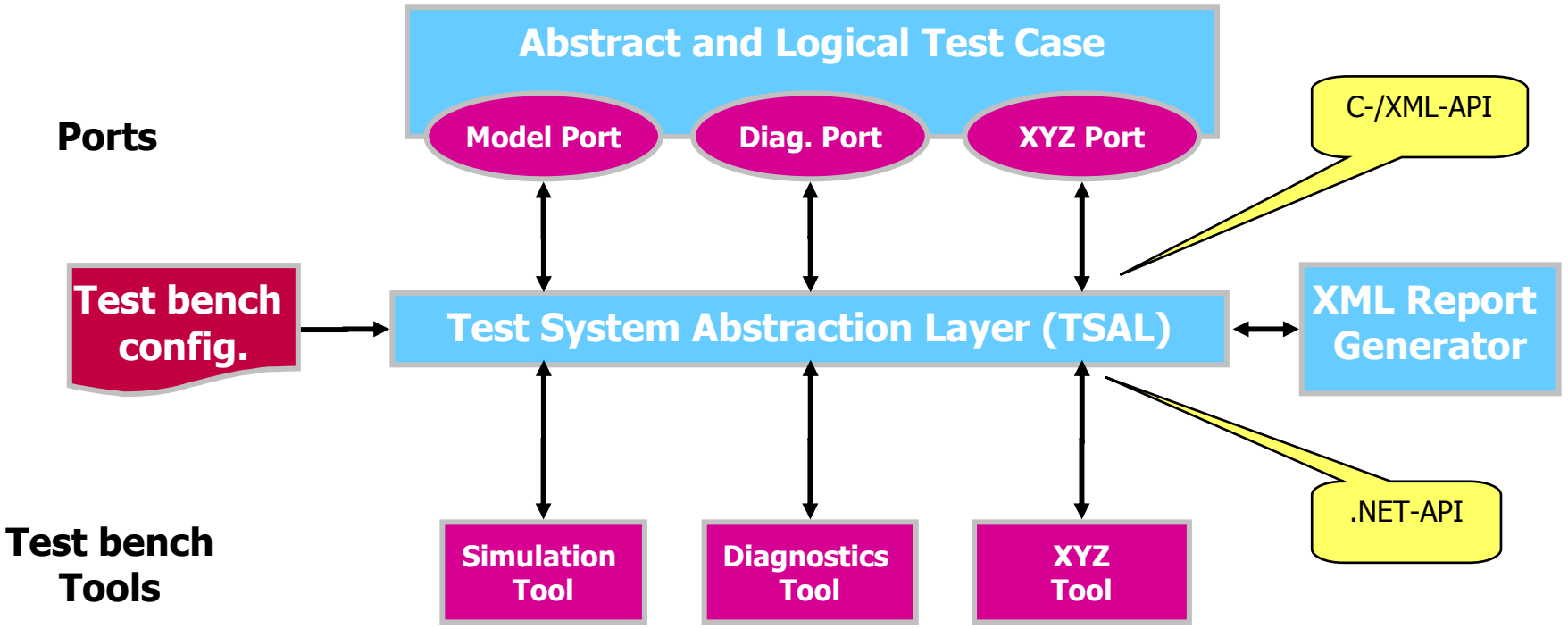
Test Execution Tools

- Test Handler
- Report Viewer
- Support for HiL, MiL, SiL scenarios
- Open APIs for flexible integration of test bench tools
- Integration package for LABCAR / ETAS test bench tools

Outline

- Evolution of Test Automation
- Systematic Approach for Test Automation Framework
- Open Interfaces
- Summary

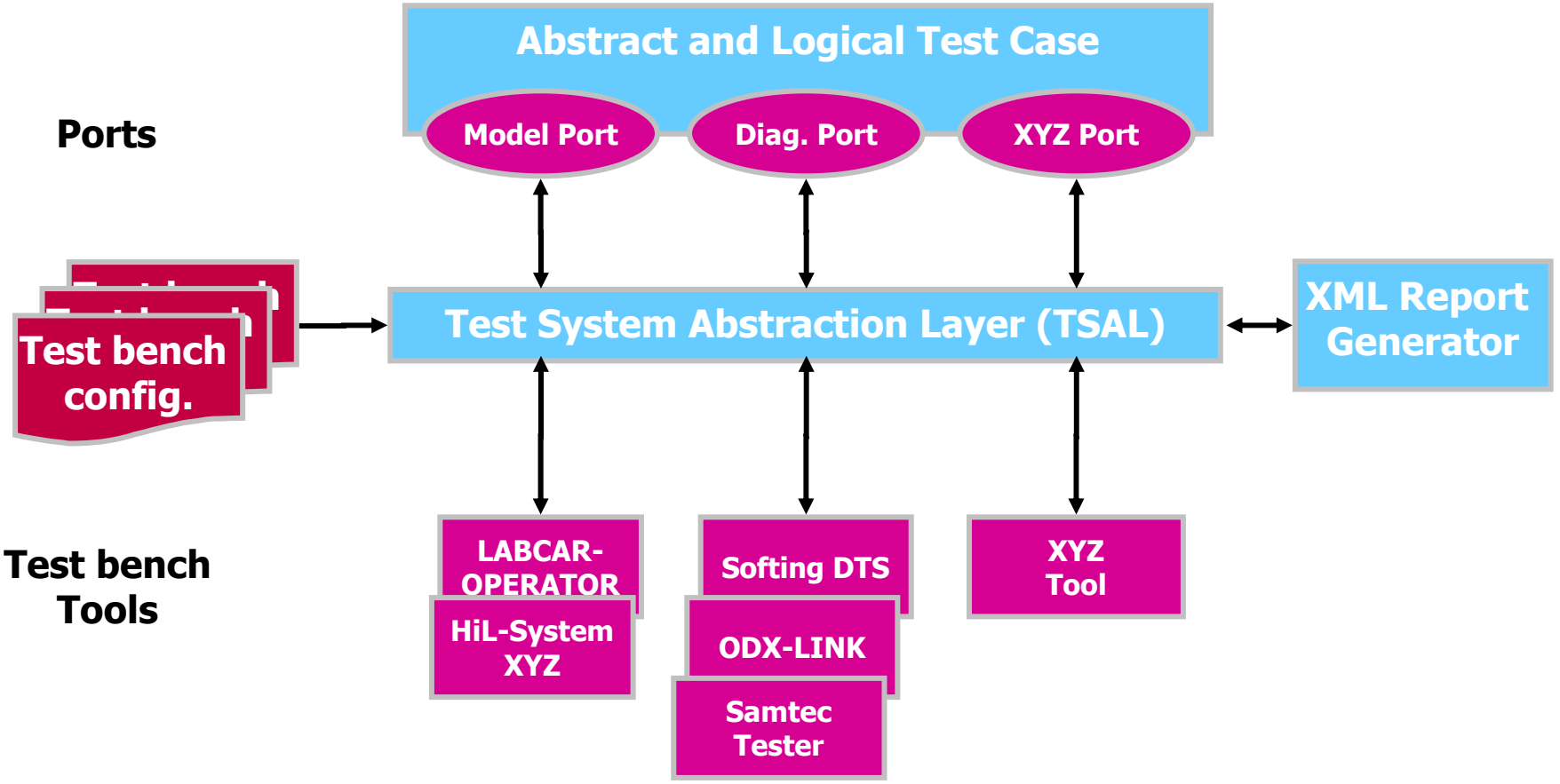
Open Interfaces



- Test-bench-independent ports for developing abstract and logical test cases
- TSAL provides open .NET API to connect to test bench tools
- TSAL provides open C-/XML-API to connect to test languages and test design tools

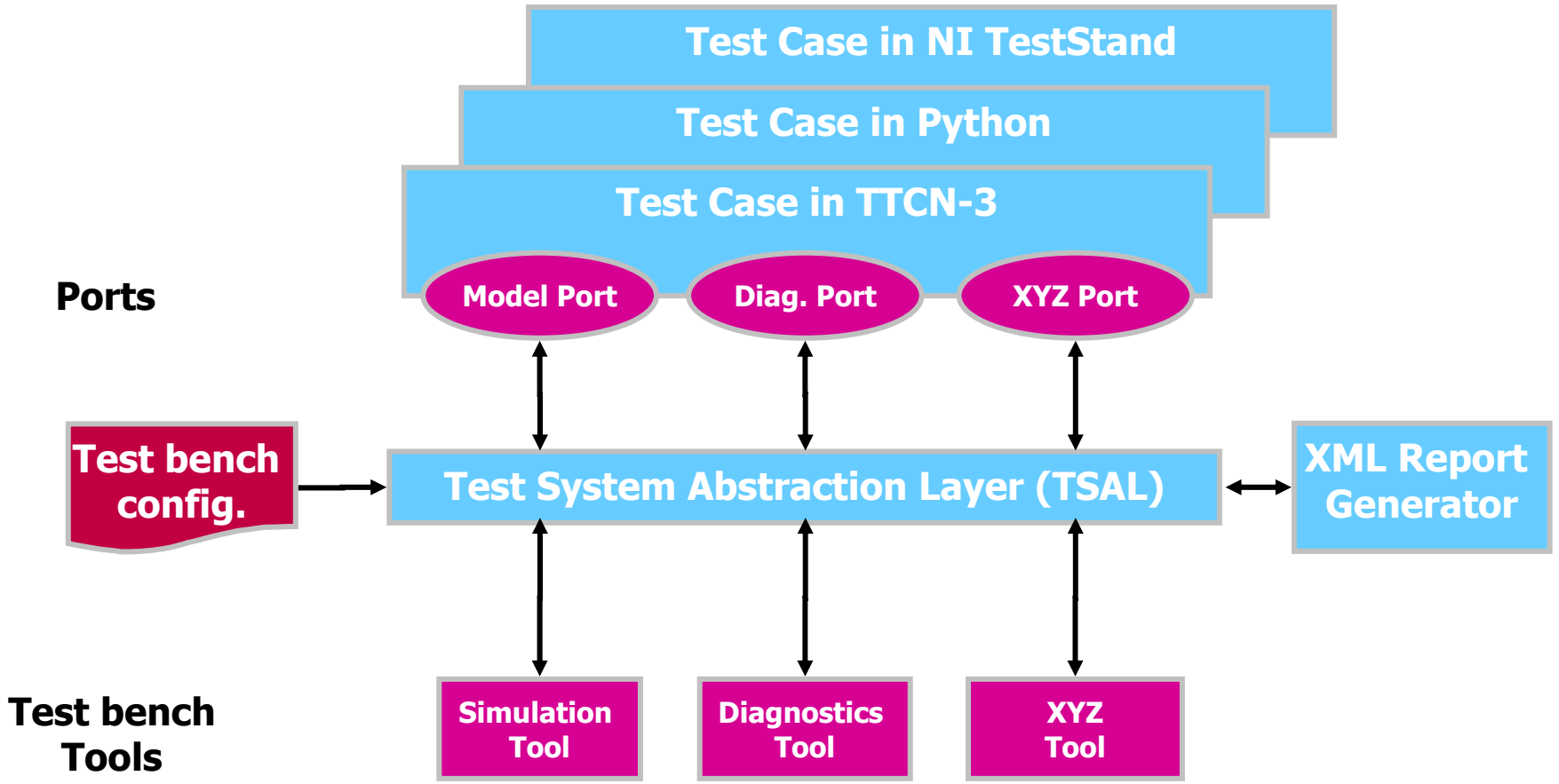
Open Interfaces

Dynamic Test Bench Configuration at Test Execution Time



Open Interfaces

Open C-/XML-API to Test Languages and Test Design Tools



Outline

- Evolution of Test Automation
- Systematic Approach for Test Automation Framework
- Open Interfaces
- Summary

Summary

LABCAR-AUTOMATION

- Graphical user front ends for
  - test case development,
  - test management and
  - test execution
  - ➢ **Process-safe test management and execution**

- Test bench- and UuT- independent test cases
  - ➢ **Improved reuse for test cases**

- Open Interfaces for
  - connection to test bench tools
  - connection to test languages and test design tools
  - ➢ **Investment protection for test case development**
  - ➢ **Investment protection for test bench tools (software + hardware)**

Thank you for your attention!

Your questions are welcome!

2005-01-1040