David Bailey, ETAS GmbH

Test And Validation: Coping With Complexity

The state of play in vehicle software and system validation

Agenda

- Test & Validation – How far have we come?
- What is driving complexity ?
- How can demands for greater complexity & reliability be reconciled?

Cars: From Yesterday to Today

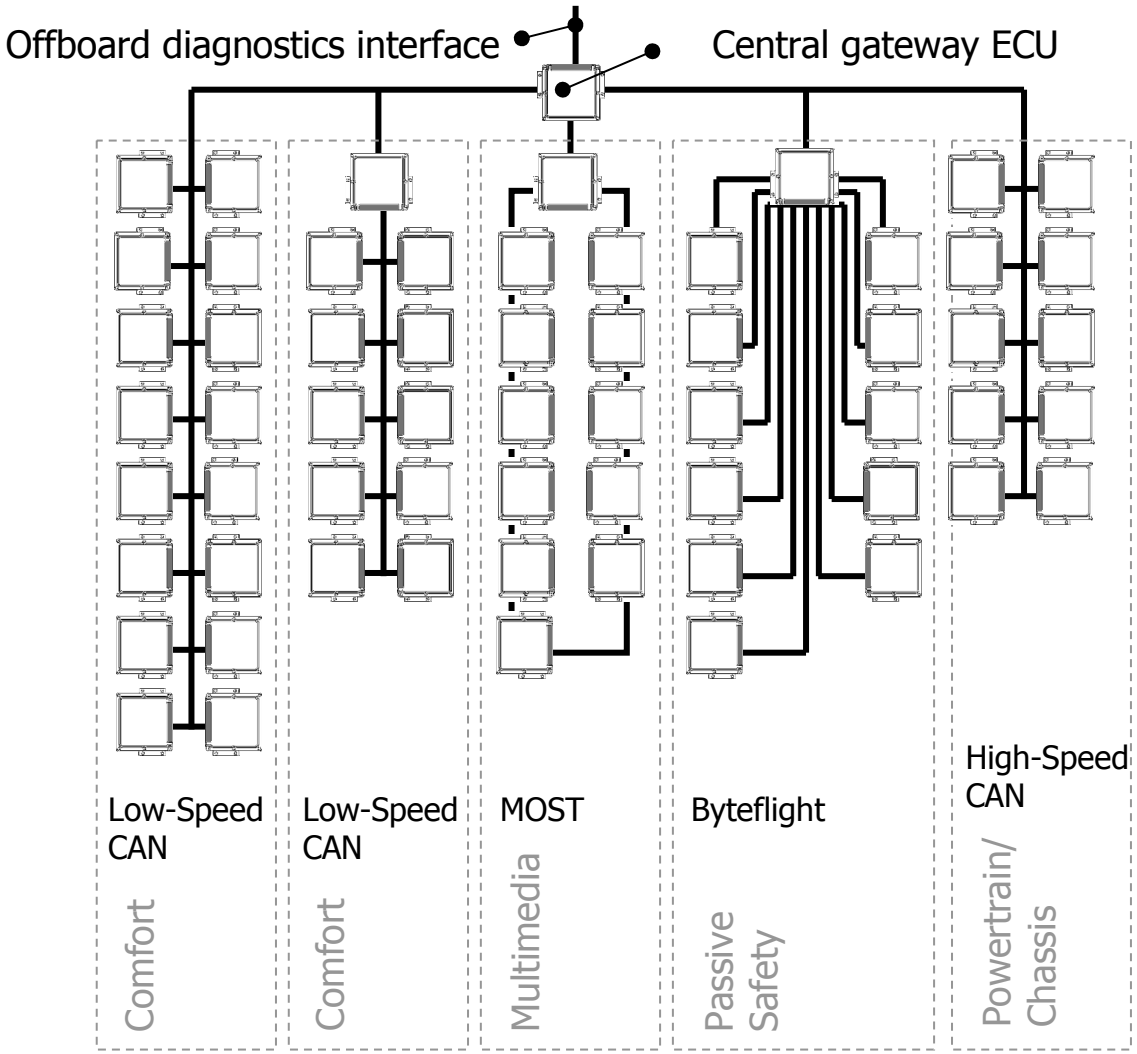
Form & Function versus Content

Wheels

Powertrain

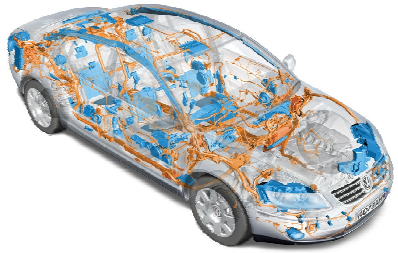
Steering

Today – Example: BMW series (E65)



- Over 65 networked ECUs
- 4 bus-systems
- ~ 116 MB code
- up to 900 functions

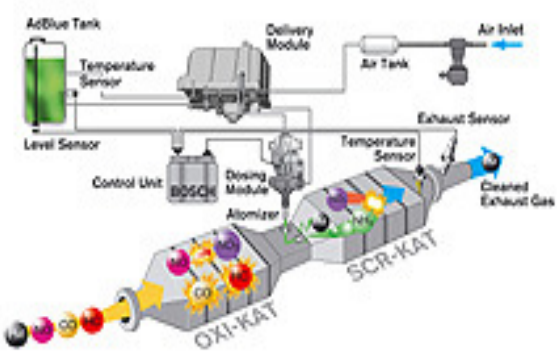
The drivers of Complexity



Warranty costs

Ever tighter emissions with safety regulations coming soon!

Increasing number of variants per platform



Increasingly complex sub-systems

Test Systems: From Yesterday to Today

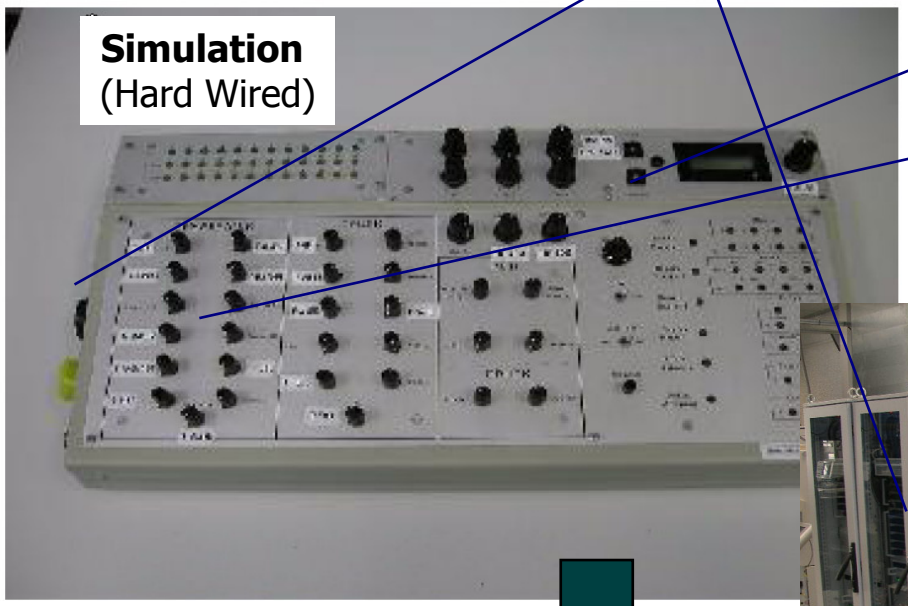
Electrical Connection
to the ECU

Computing nodes

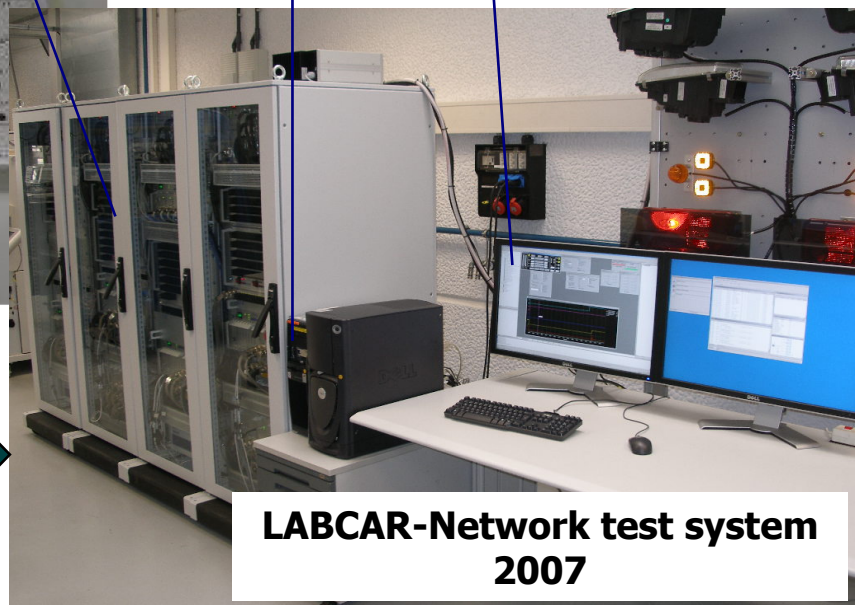
User Interface

Automation

Virtual
Environments



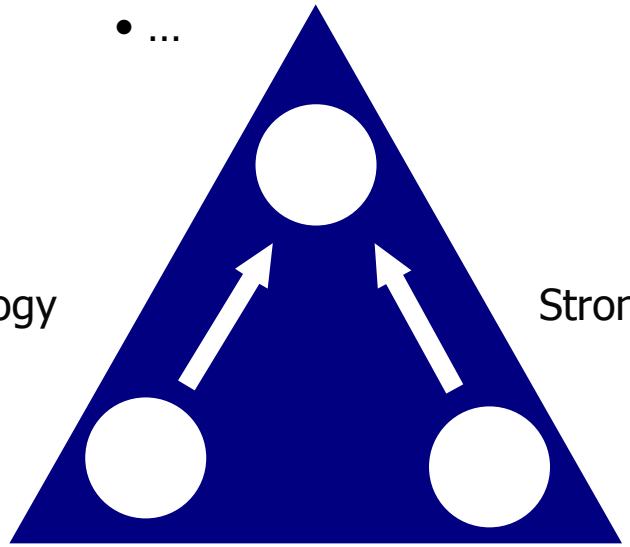
Which drivers have driven
this developments?



LABCAR-Network test system 2007

Drivers in 3 Dimensions

Application
- Test-processes
- Test System Configuration
- Test Execution
- …



Yesterday
Focus on Technology

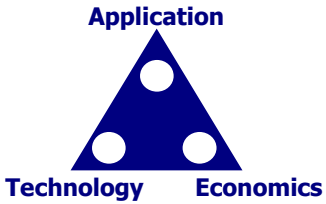
Today
Strong focus on economics – Process-orientation

Technology
- Increasing use of PC technology and Open Buses
- Open Models
- Open run-time environments

Economics
- Cost-Benefit Relationship
- Cost over lifetime
- Re-usability
- Process integration

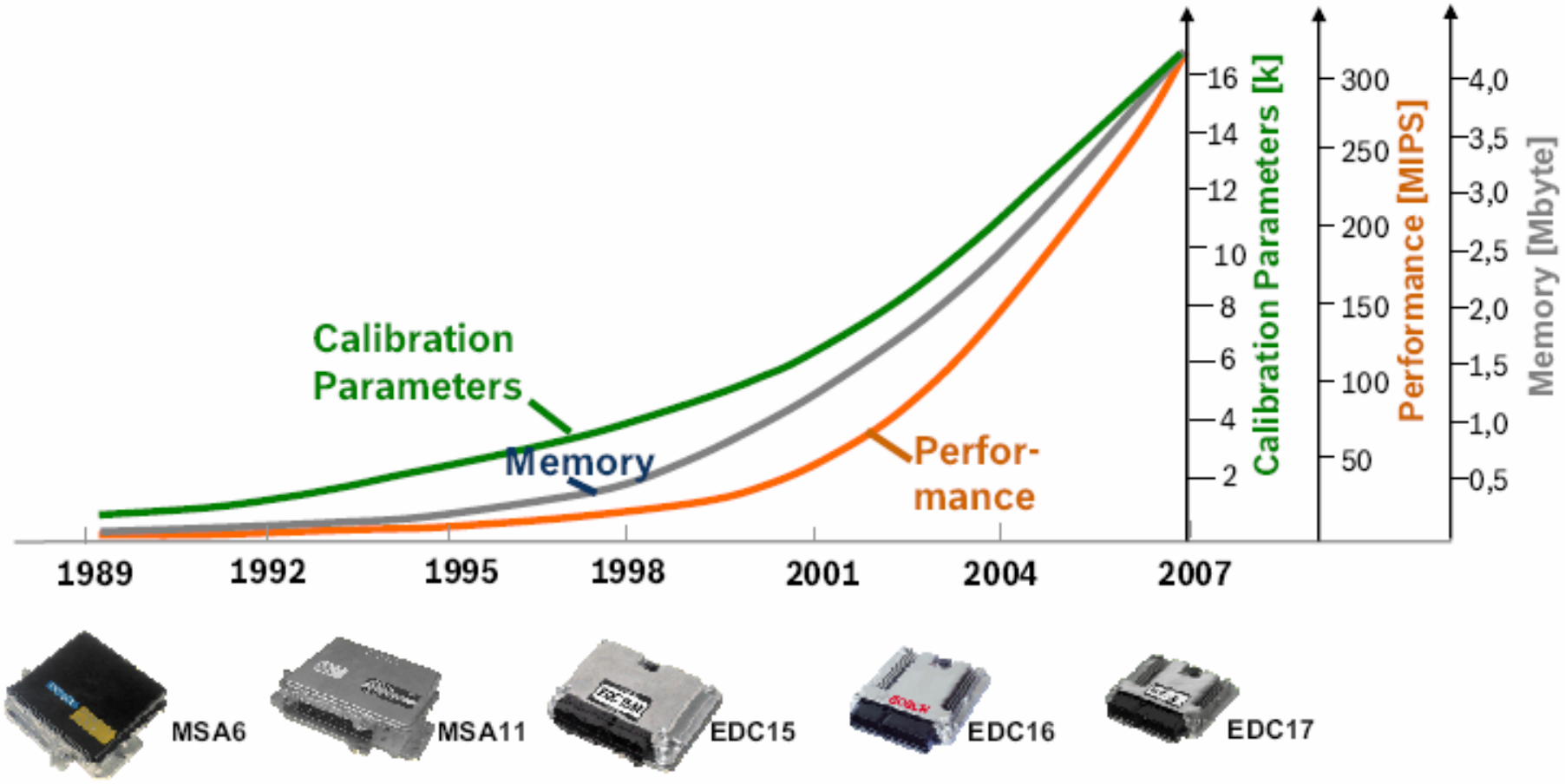
Test & Validation: from Yesterday to Today

Trends and consequences



Trends		Yesterday	Today
Increasing ECU functionality (Quantitive & Qualitative)		• Manual Testing • Simple Models • Slow real-time micros	• Automated Testing • Complex Models • Fast real-time micros
Increasing Networking (Quantitive & Qualitative			
Sinkendes Budget			

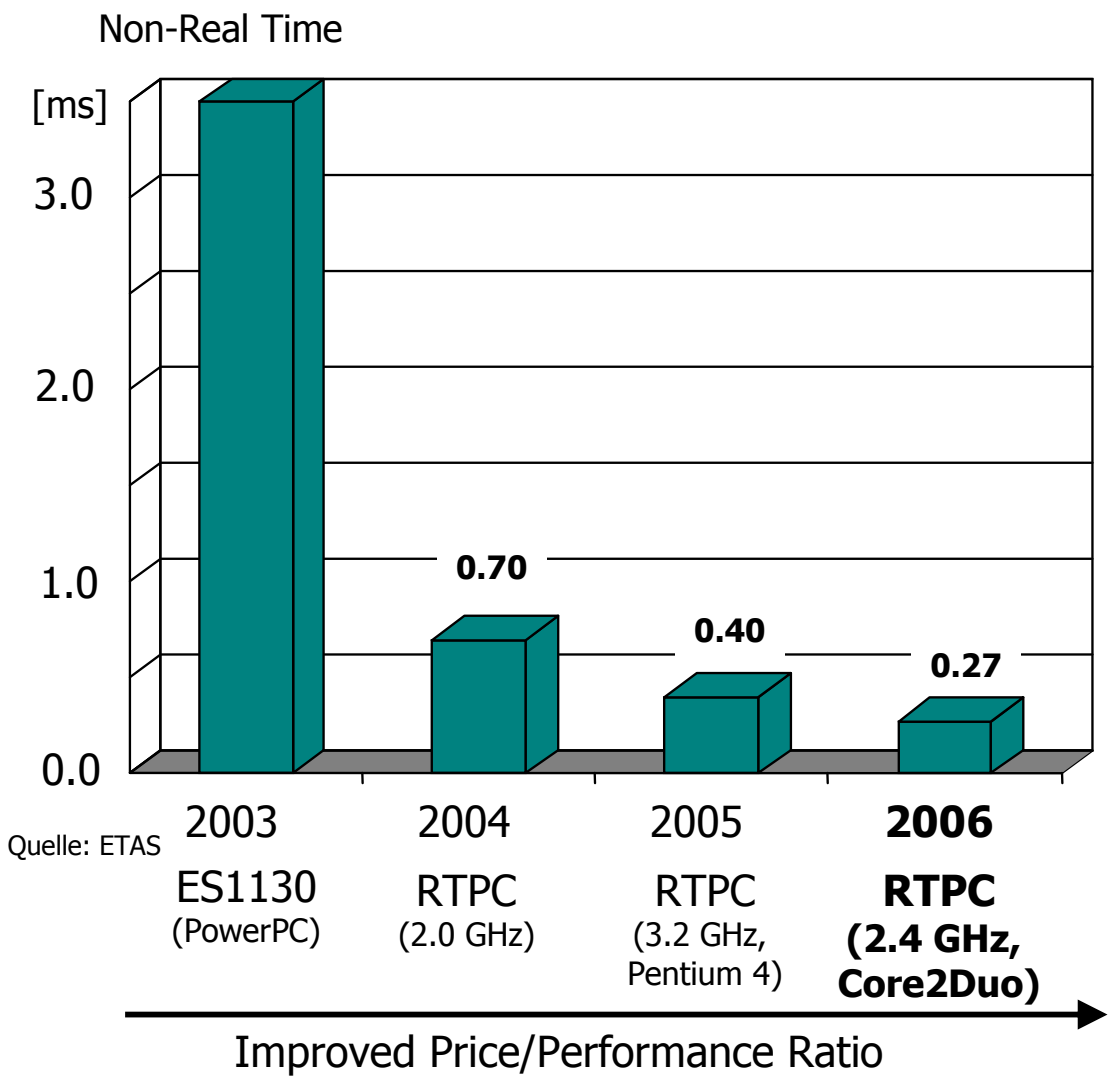
Complexity growth (Diesel Systems)



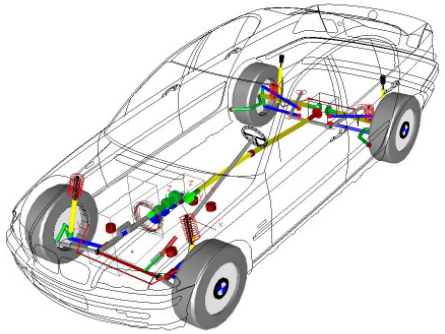
Source: Bosch Diesel Systems 2007; DS/PJ-EER; EEH0 07 024

Increasing ECU functionality: Fast real-time µs replace slow ones

Technology



Quelle: ETAS



Calculation time [ms]

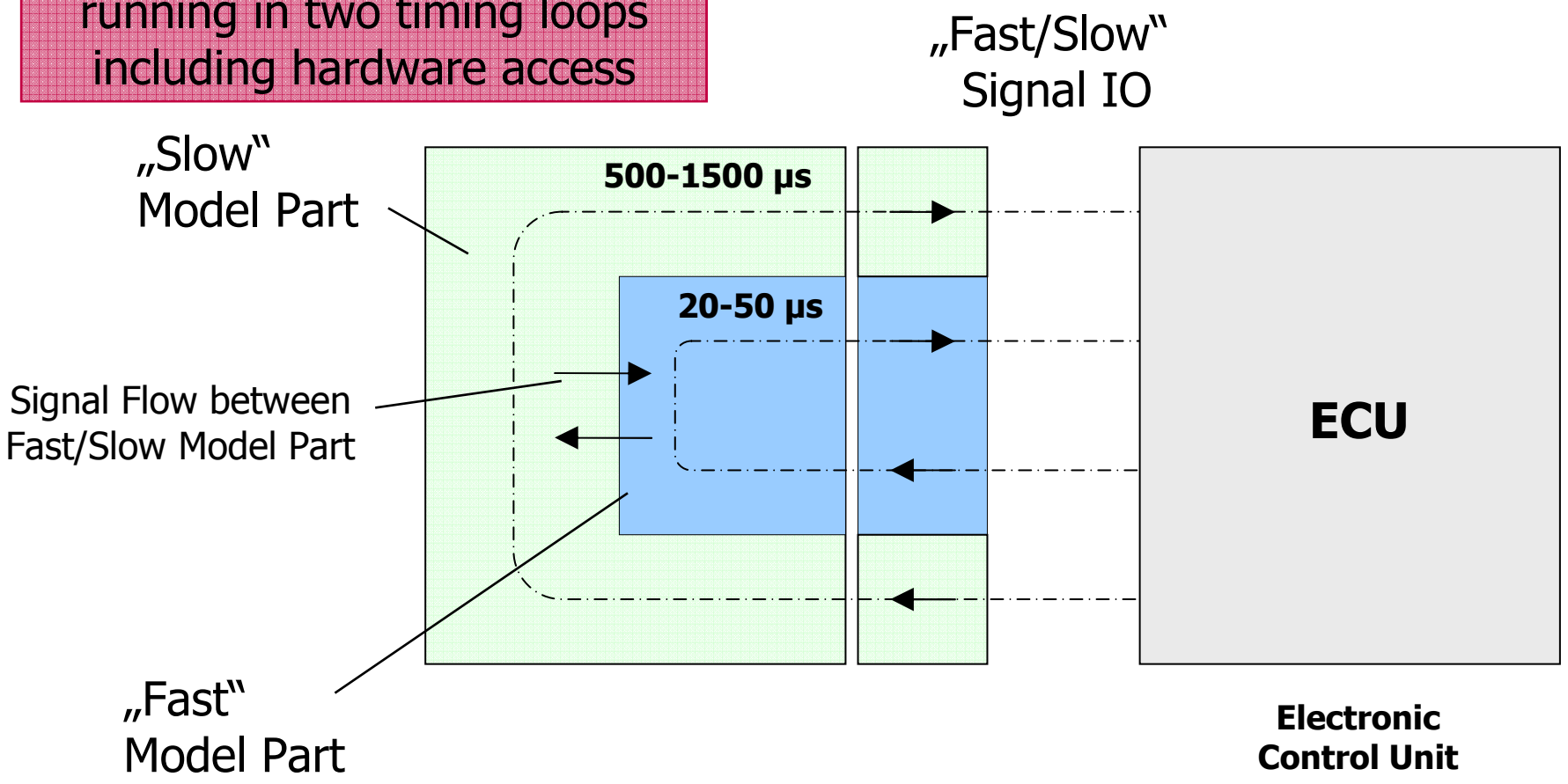
- Euler-integrations
- Vehicle Dynamic Model LABCAR-VDYM V5.0
- Excludes I/O turnaround

Multi-Core Processors – "Triggered Sub-Systems"

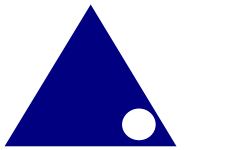
Benefits of multi-core processors in model execution

Technology

Real-time model
running in two timing loops
including hardware access



Embedded Software Testing Requirements

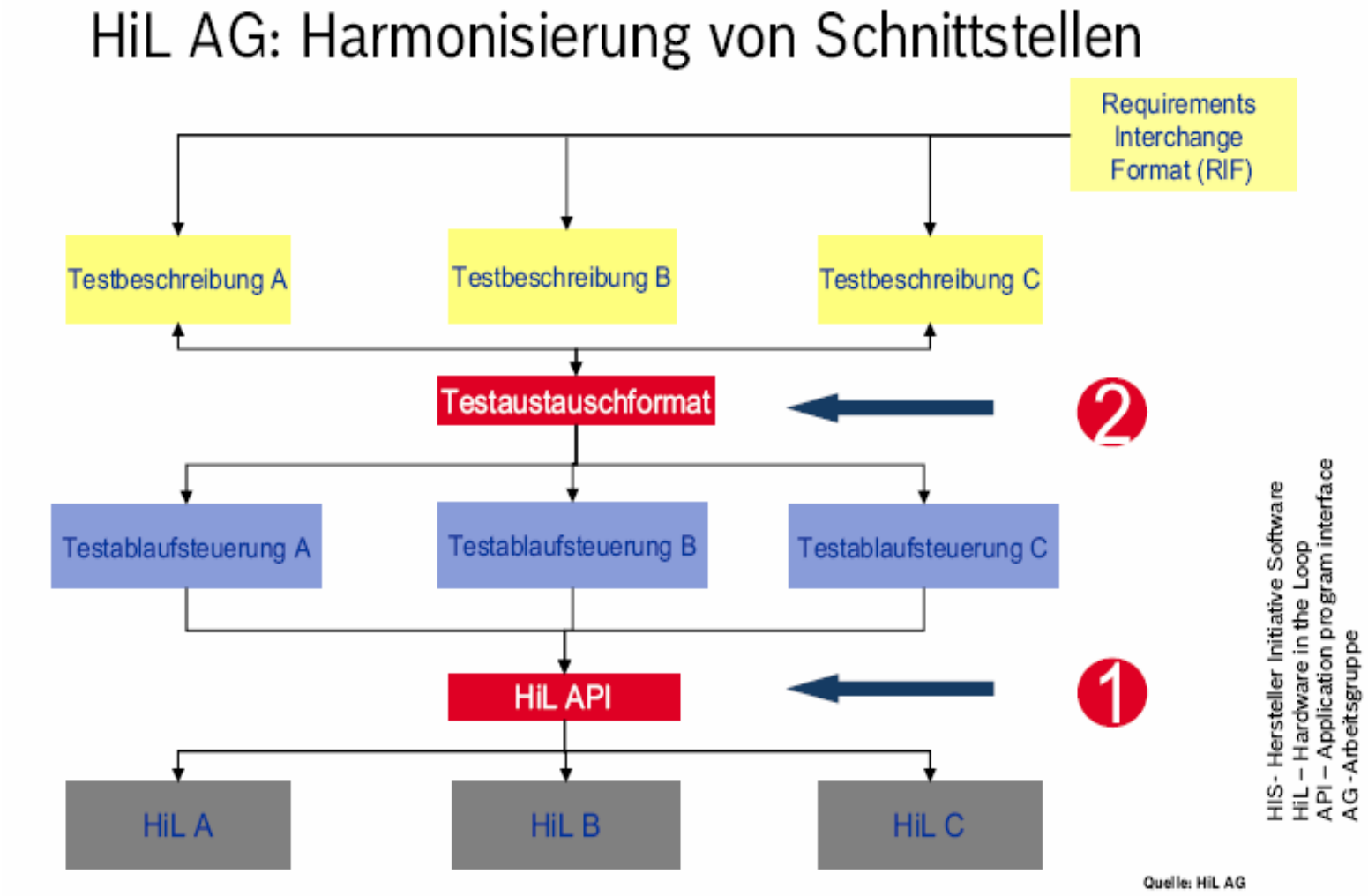


TASK: Increase test cost & coverage with lower budgets

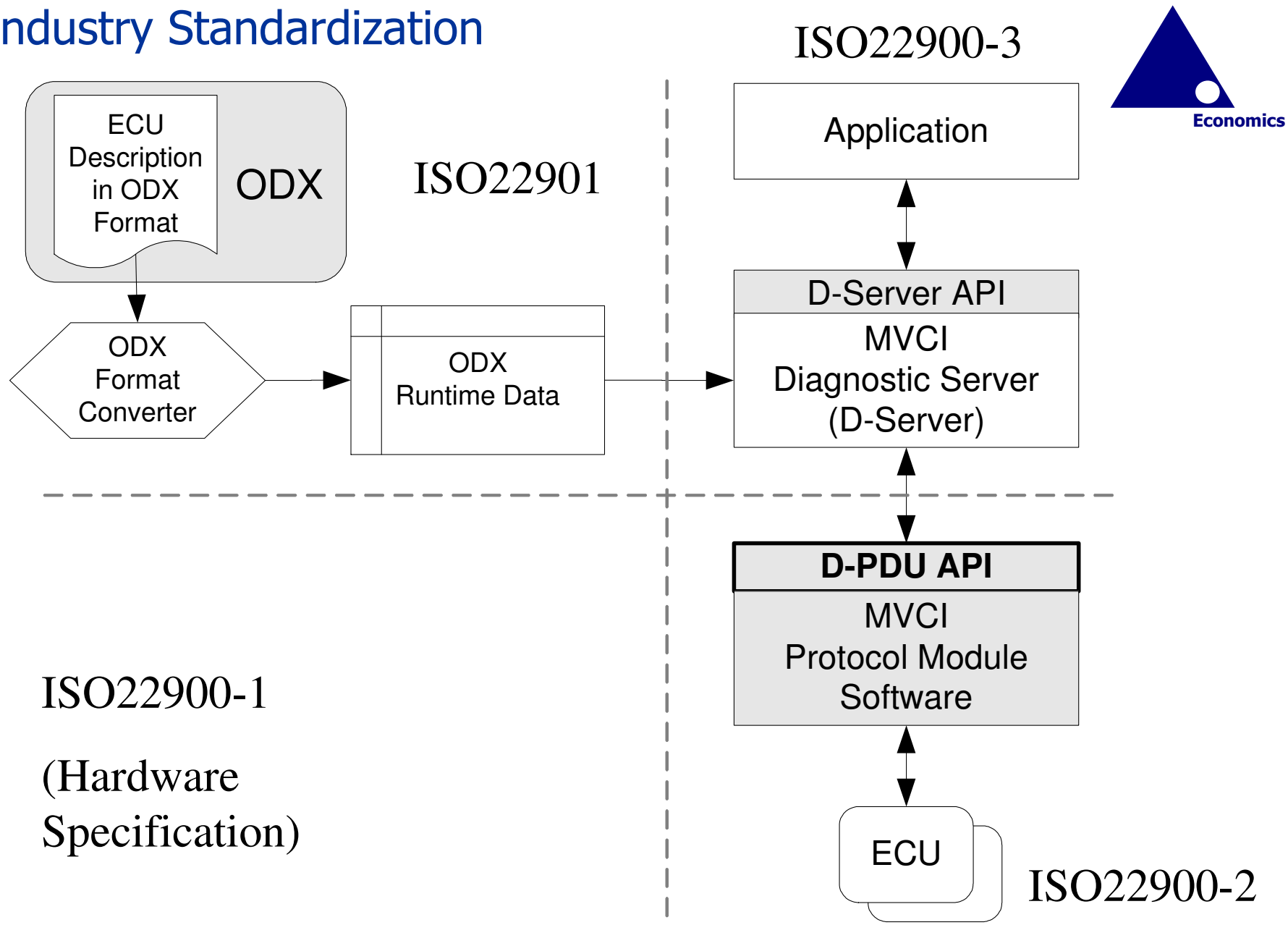
Requirement	Advantages & Characteristics
Test Stand Independence	➢ Guaranteed re-usability of test-cases between projects, systems, versions and variants ➢ Protection of Intellectual Property ➢ Reduction of implementation effort (once, for ever)
Independence from Development language	➢ Reduction of learning effort ➢ Re-usable development environment
Parametrised (logical) test cases	➢ Simple but powerful adaptations possible for specific test applications ➢ Limiting the development effort "one test-case per specification" ➢ Lower the administration effort
Test Project Management	➢ Management of different combinations of test cases with parameter-sets with minimal effort
Integration in Test-processes	➢ Increase Transparency & Traceability ➢ Open interfaces (eg to requirements management tools (Doors, SVN…) ➢ Dedicated roles (There is no universal test-environment!)

Industry Standardization

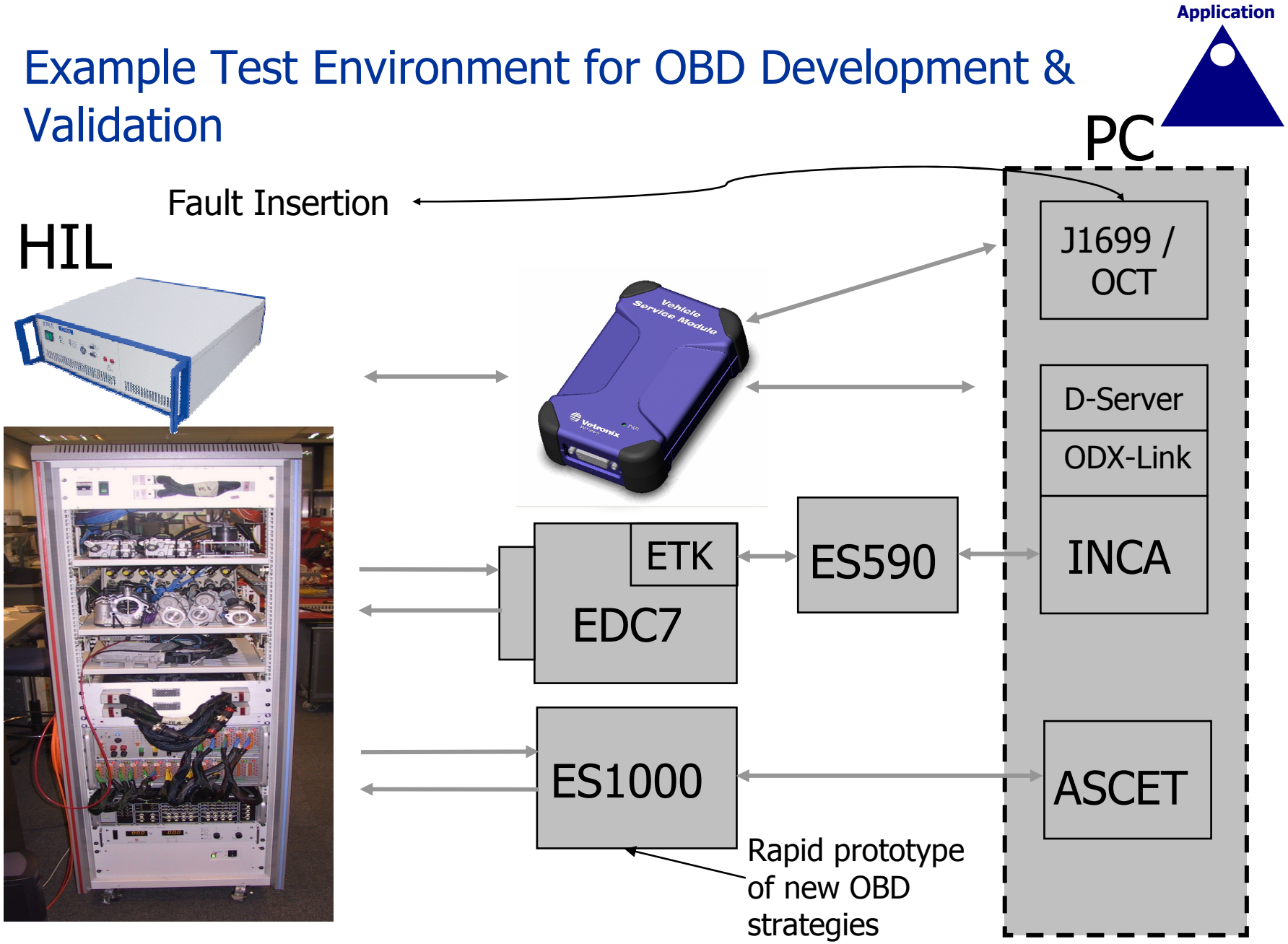
Economics



Industry Standardization



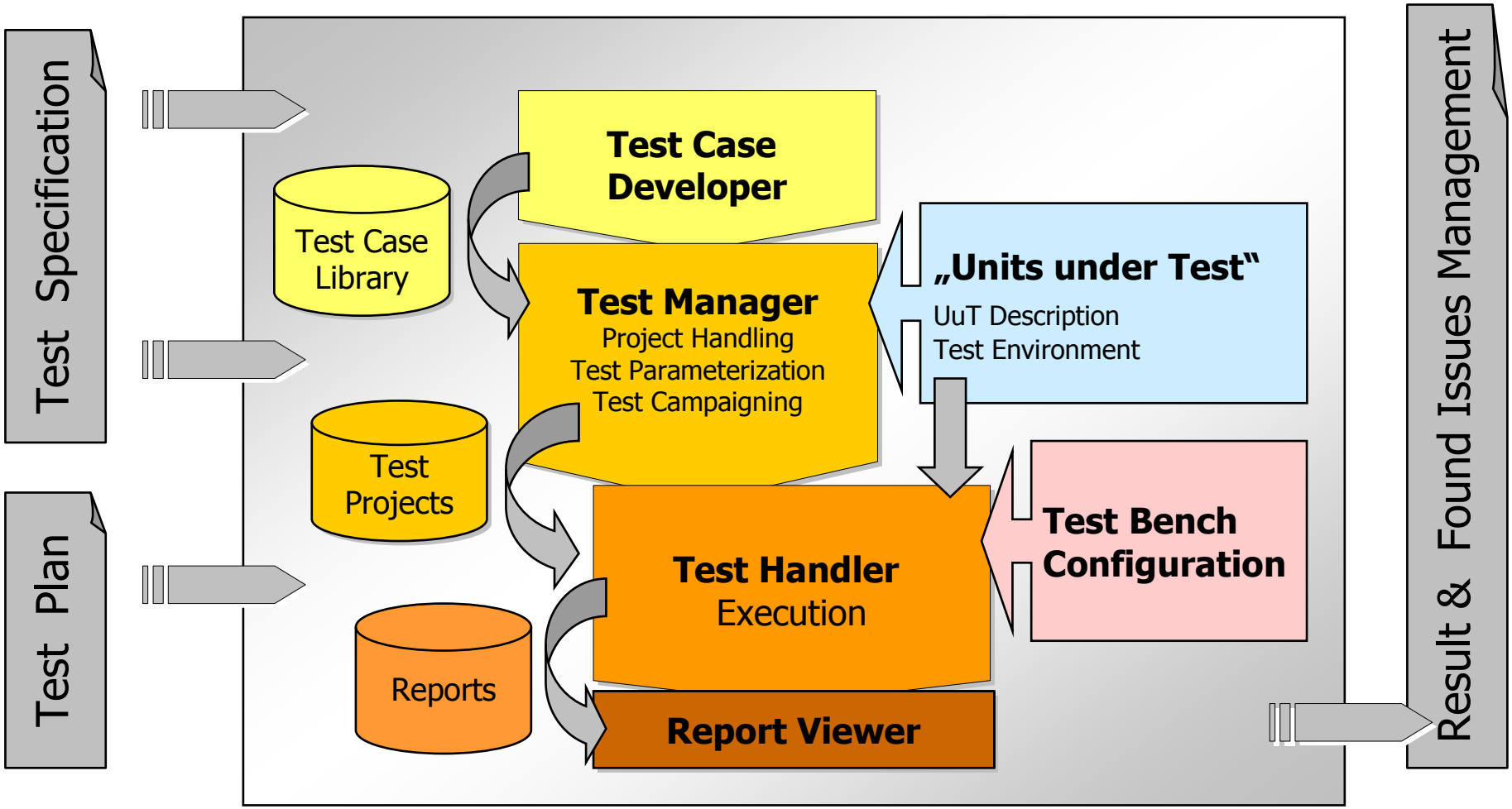
Example Test Environment for OBD Development & Validation



Description and allocation of roles

Test automation completes or replaces manual testing

Application



Requirements of the Test Developer

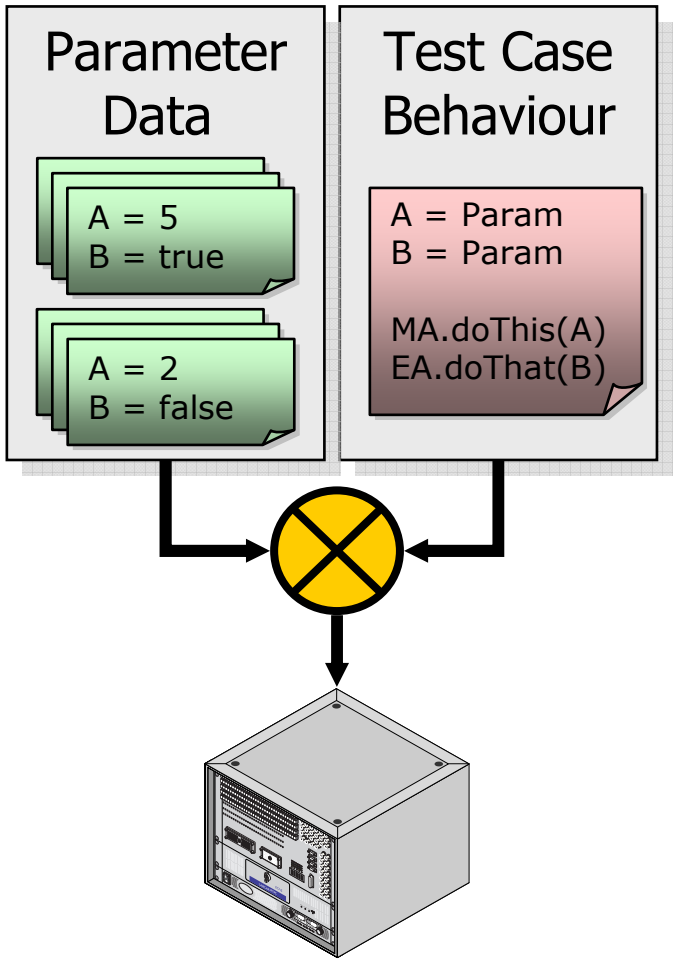
There is no universal test language!

Language Type	Example	Comment
Test – oriented	TTCN-3	➢ Rather complex ➢ Has specific Test constructs (e.g. "Verdicts") ➢ Small and specialized user circle
Script	Python, Perl, m	➢ Little learning effort ➢ Limited Process security (no compiled test-cases) ➢ Normally not "strong typed" (a big problem for operational consistency)
Graphical	Simulink, UML tools, NI TestStand	➢ User friendly with guided operations ➢ Guaranteed syntactical correctness ➢ Sub-optimal for sequential test architectures ➢ Possible loss of oversight with big projects ➢ Lack of transparency
Software Development	C#, VB, C++	➢ Very powerful development environment (also low cost) ➢ Large user group ➢ Higher learning effort ➢ No specific Test Constructs (e.g. verdicts)

Example: Test Parameterization

Fundamental for re-usability

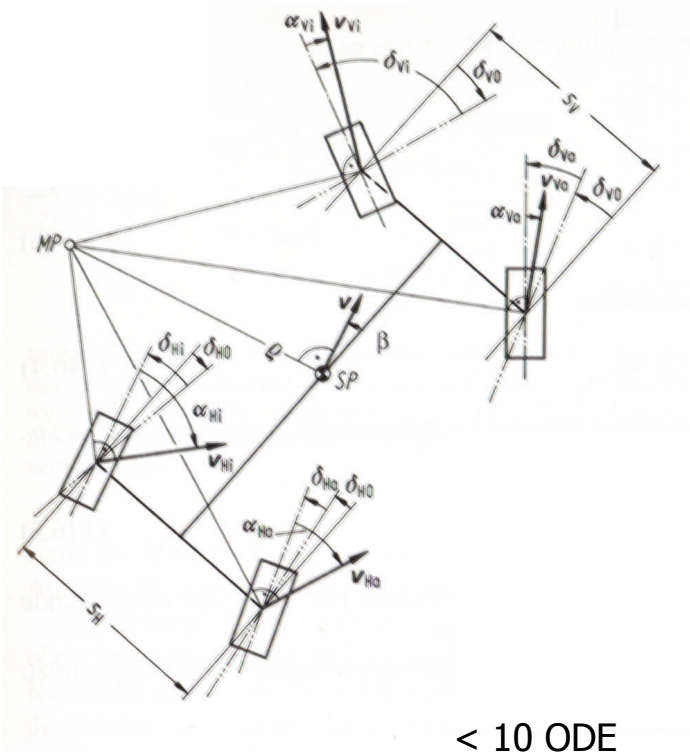
Use of the parameter	Description
Adaptation of **Test Systems**	Data for specific Test-system configurations, time-outs & tool options.
Setting and definition of the working point of the tests	Environmental Data, Fine-tuning of the plant-models to the test conditions (i.e. dependent on the Unit under Test)
Setting the **Test-points** and **Test -vectors**	Describing the Test scenario. (e.g, arc diameter, entrance speed, exist speed, braking force… for ESP test)
Settings for **Evaluation Method**	e.g. Selection of method and definition of pass/fail thresholds



Increasing ECU functionality

Complex Real-Time models replace simple RT models

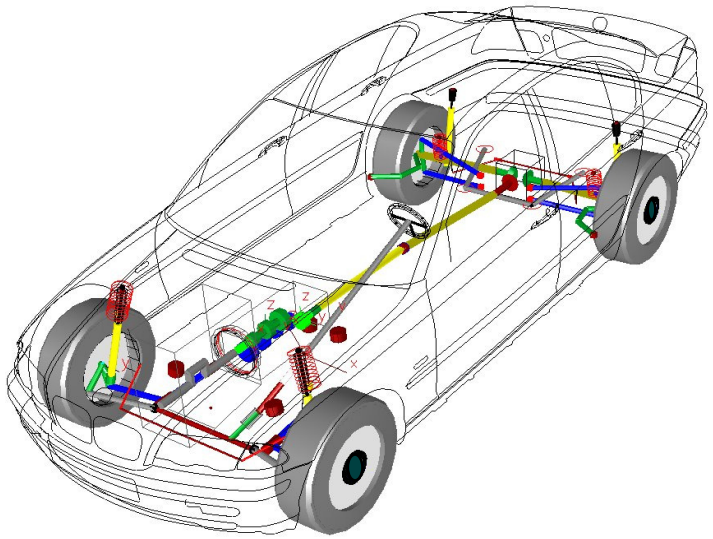
Technology



< 10 ODE

Yesterday: Simple two-track model for Vehicle Dynamics

(Source: Dynamik der Kraftfahrzeuge, Mitschke)



< 150 ODE

Today: Complex MBS-vehicle dynamics model with Axel-geometry

(Source: INTEC, LABCAR-VDYM V5.0)

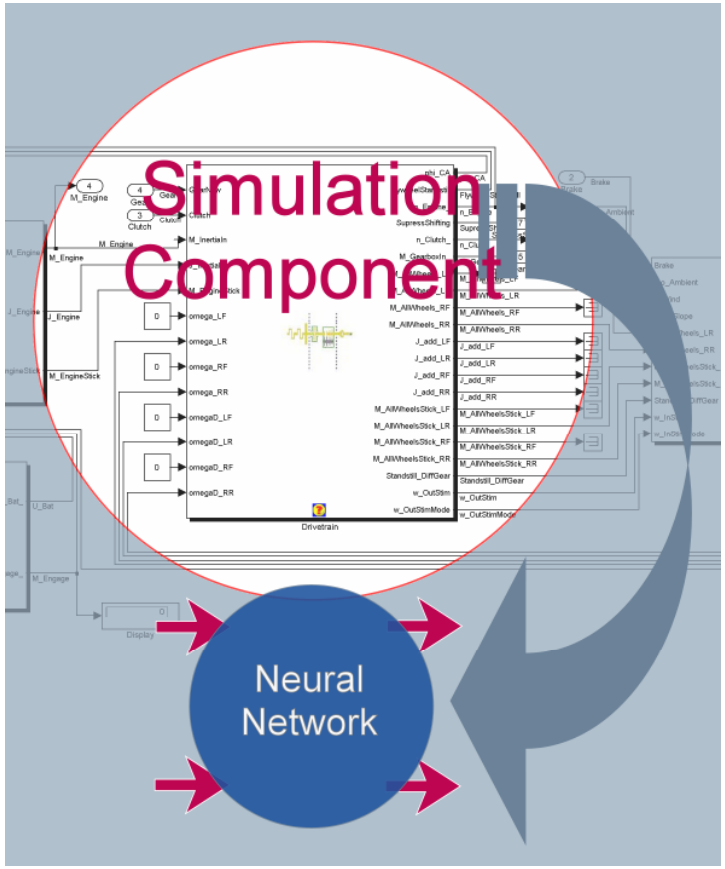
ODE = Ordinary Differential Equations, **MBS** = Multi-body simulation

Model Optimization using both physical & statistical, data driven methods *(e.g.TLRNNs, SOM/TFA/LLM)*

Technology

The Problem

- There is a break in the tool-chain between CAE tools used for basic powertrain design and plant models used for controls development
- How to assess design impact on emissions?
- How can complex models still run in real time?

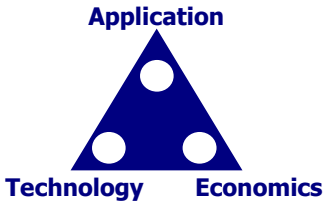


Solutions

- Complex „logical" optimization of the physical model *(eg:WAVE RT von Ricardo)*
- Statistical Modelling Tools *(Gamma Technologies, The Mathworks)*
- Engineering solutions.

HiL-Simulation: from Yesterday to Today

Trends and consequences

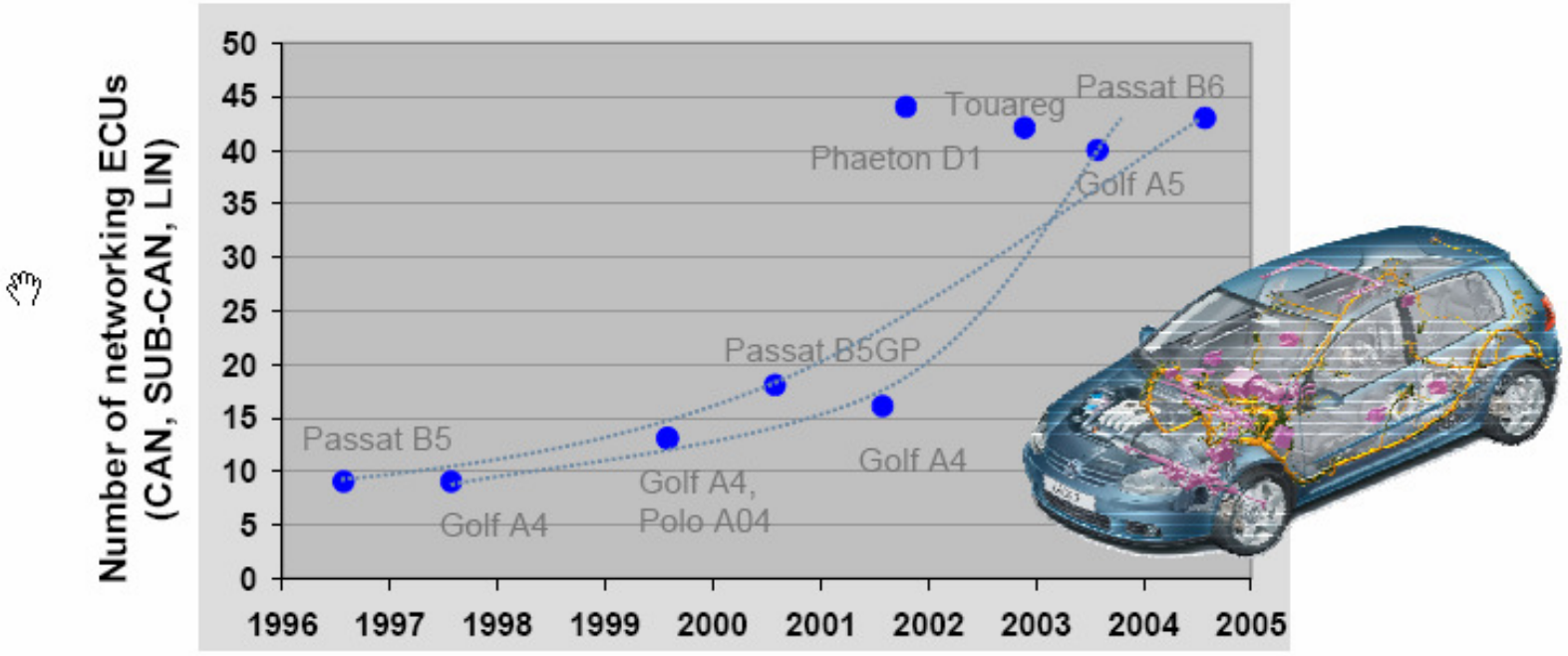


Trends		Yesterday	Today
Increasing ECU functionality (Quantitive & Qualitative)			
Increasing Networking (Quantitive & Qualitative)		• Single HiL Systems • Slow Bus-systems • Sequential Development process	• Networked HiL Systems • Fast Bus-Systems • Iterative Development Process
Sinking Budget Budget			

Increasing Networking

Indicators

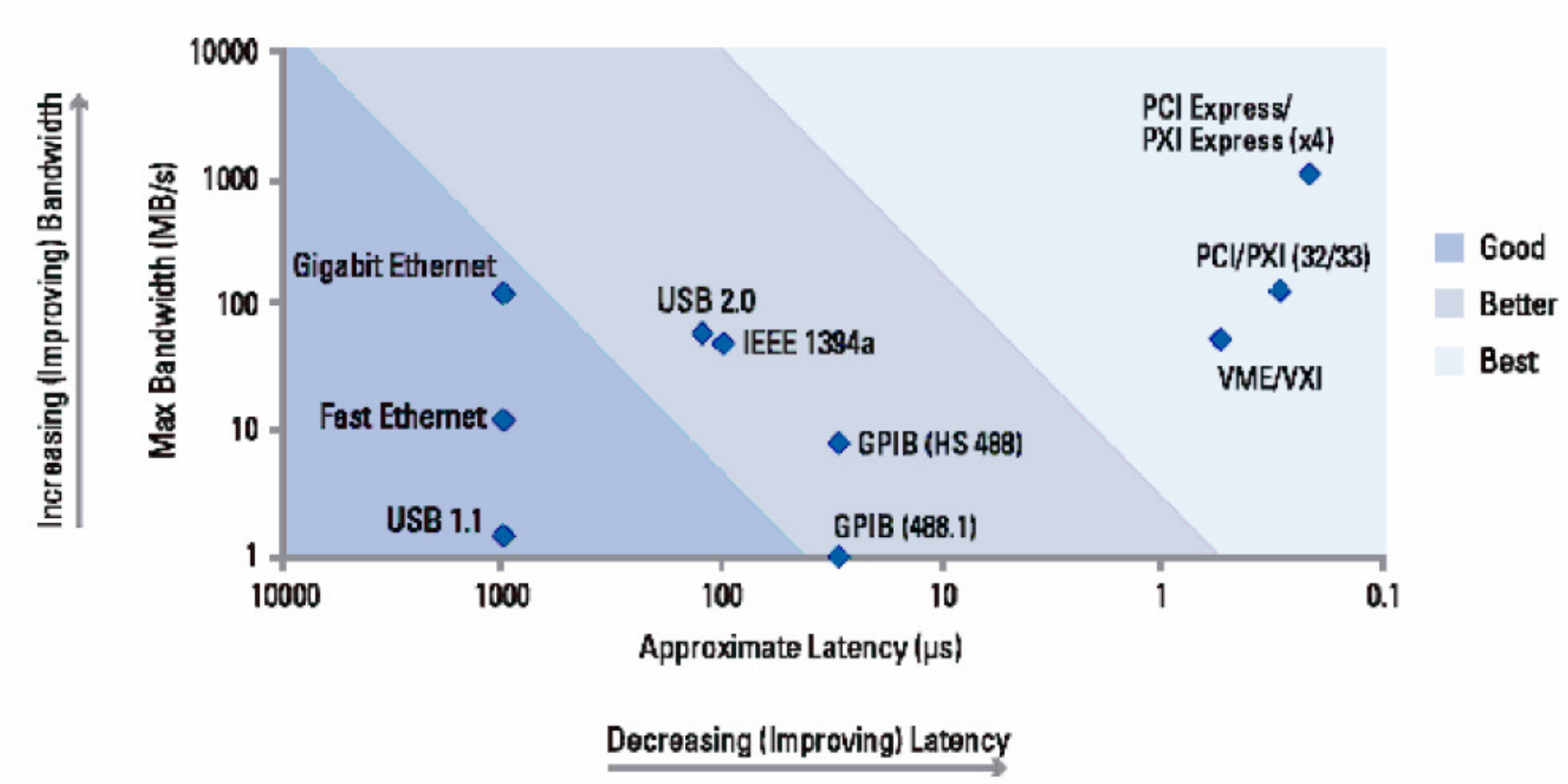
Technology



Quelle: Volkswagen AG (Grafik)

Increased Networking – HiL- & PC worlds
Fast bus systems replace slow ones

Technologie

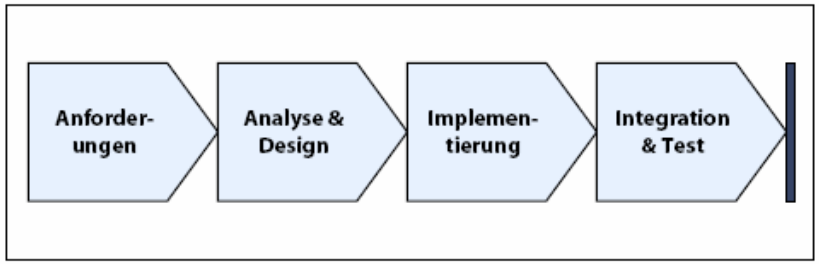


Quelle: www.ni.com, National Instruments
White Paper - Bus Performance.pdf

Increasing Networking

Iterative Processes replace Sequential ones

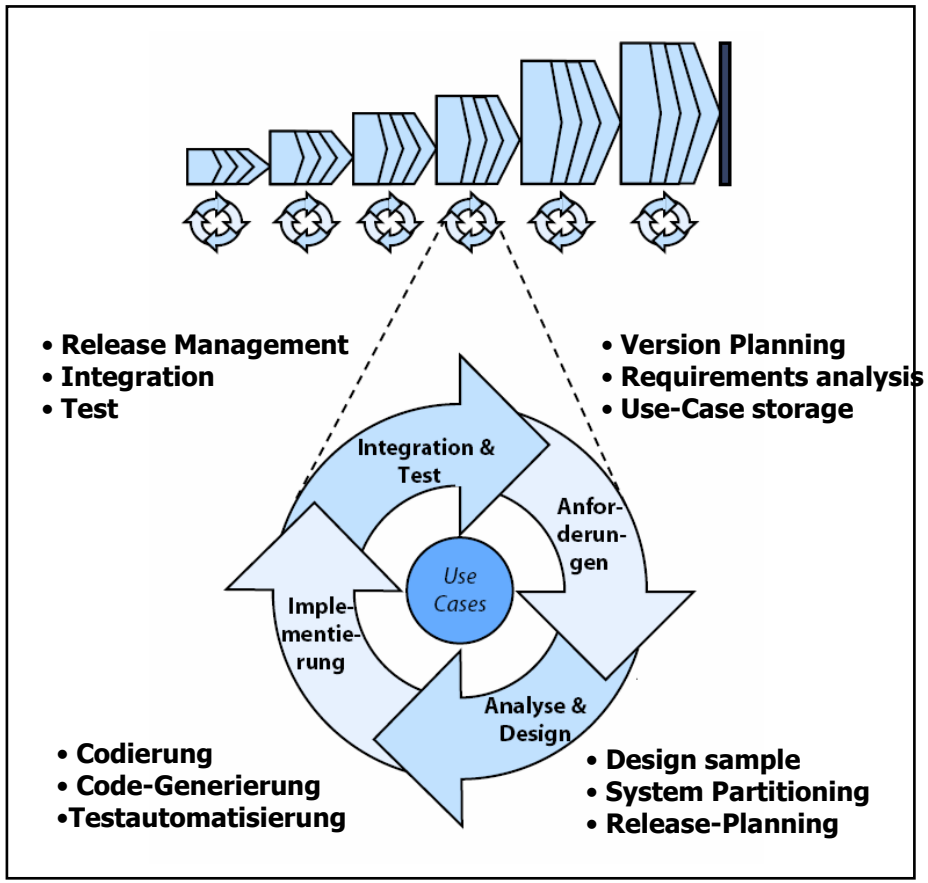
Economics



Quelle: www.basycon.com
POSTER_SW-Qual_EntwProz.pdf

Vehicle Development is ‚largely' software development

→ The Development process should adapt accordingly

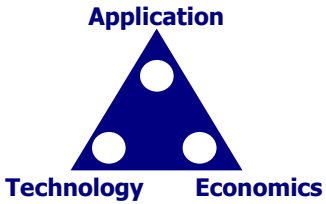


Yesterday: Sequential Development Proces

Today: Iterative Development process.
Increasing coherence MiL-SiL-HiL

HiL-Simulation: From Yesterday to Today

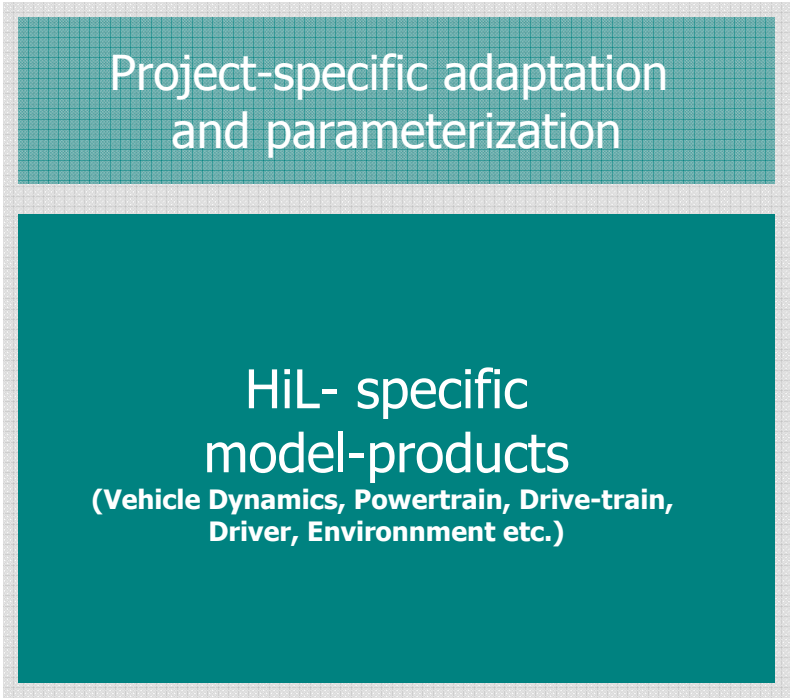
Trends & Consequences



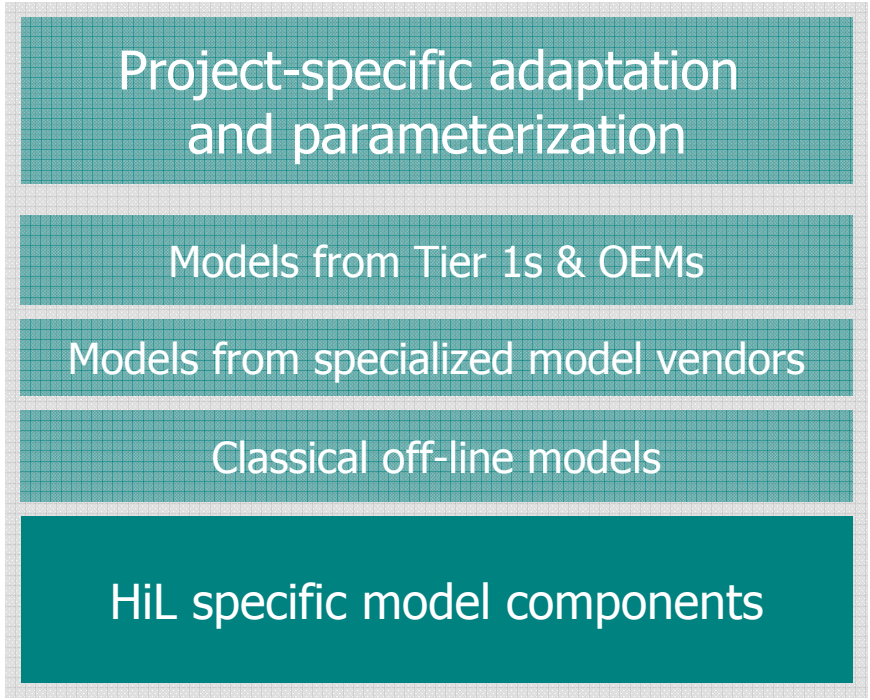
Trends		Yesterday	Today
Increasing ECU functionality (Quantitive & Qualitative)			
Increasing Networking (Quantitive & Qualitative)			
Sinking Budget Budget		• System-based service • Proprietary models • Technology-driven product solution	• Solution-based service • Open models • Application driven Ppoducts • Re-usable test-cases • Outsourcing of test activity

Sinking Budget, rising requirements

Open model solutions integrate with complex models



Yesterday: HiL-specific real-time models

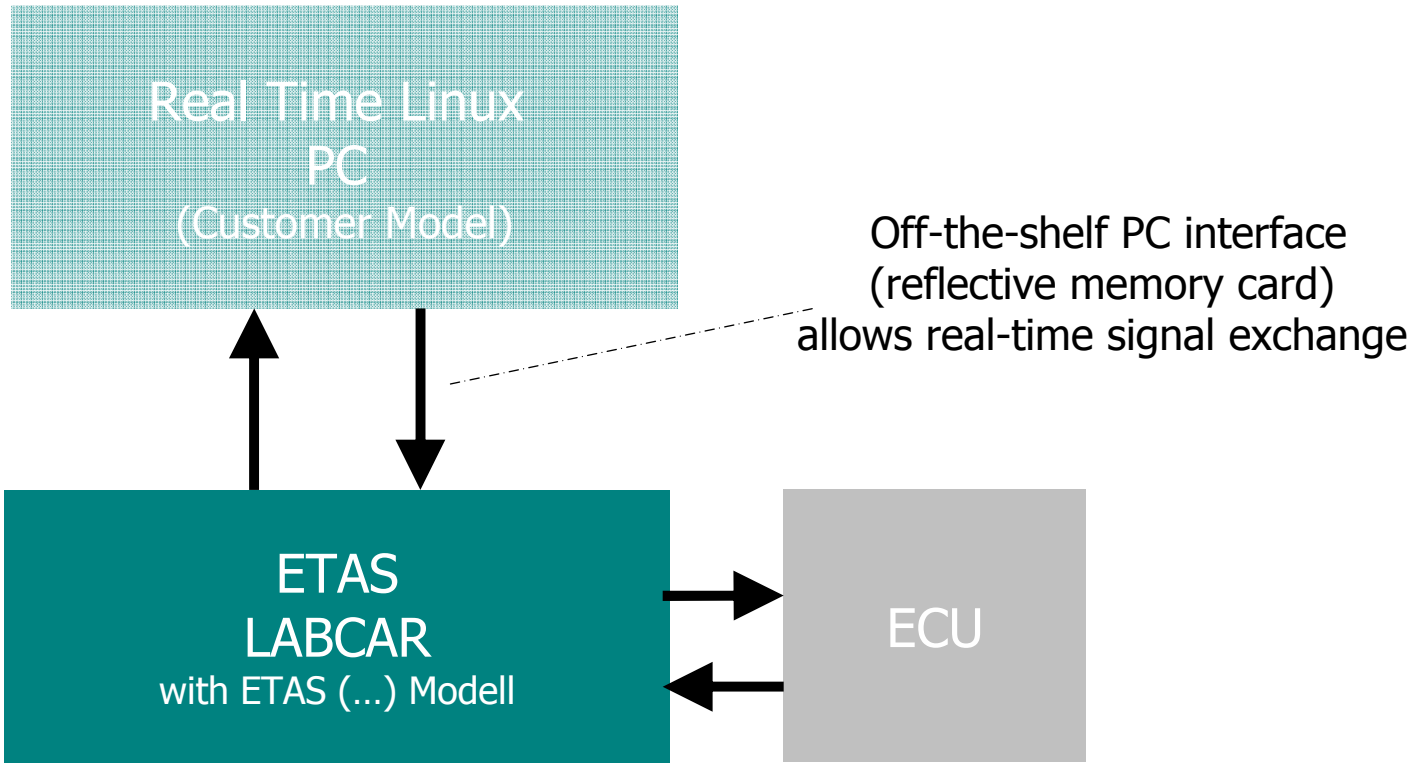


Today: HiL specific models extended with a variety of specialized models from different sources

Sinking Budget – Rising Requirements

Open integration example

Technology Economics



Sinking Budget

Application-driven solutions replace technology-driven ones

Economics

	Power-Train	Chassis	Body	Others…
 • High-End HiL-System				
 • Standard HiL-System	**Yesterday:** Technology Driven (z.B. ETAS *VME-LABCAR*)			
 • Basic HiL-System				

Sinking Budget

Application-driven solutions replace technology-driven ones

Economics

	Power-Train	Chassis	Body	Others…
• High-End HiL-System	Today Application Driven e.g. ETAS *PT-LABCAR*			
• Standard HiL-System				
• Basic HiL-System				

What will the future bring?

- **Application**
  - New Applications — Increasingly „virtual" testing; Reusability & Interoperability; Common Test Environments deployed at new stages of development
  - Standardization — Test languages, Tool APIs
  - Automation — Increasing Test Automation, also other development processes automation (e.g. calibration)

What will the future bring?

- **Technology**
  - Increasing Processing Power — DualCore, QuadCore, …
  - Faster PC Buses — PCI-Express, US
  - Virtualization — Increasingly high fidelity,real-time capable models
    Increasing model Types (e.g. Processor, ECU models, network models)

What will the future bring?

- **Economics**
  - Further Efficiency Drives — 24/7 „Test Houses"
  - Further Cost Pressure — Road → to Lab → to Math
  - Outsourcing — Reducing costs but driving requirement for MUCH improved process security
  - Standardization — Further industry wide standardization moves
    Increased utilization of de facto standard technology (PC, .NET…)

Thanks for your attention!

David Bailey
Business Development Manager – ETAS GmbH
+49 711 89661 371
David.bailey@etas.com
www.etas.com

David is responsible for business development at ETAS for Test & Validation solutions. He has been working at ETAS for 4 years. Previously David has worked for Dearborn Group Inc where aside from business management in Europe he participated in a number of standardisation committees related to vehicle bus protocols & ECU reprogramming.