

## Decoupling Test Cases from Real and Virtual Test Systems with ASAM HIL API



Dr. Rainer Rasche, dSPACE GmbH

Dr. Dietmar Neumerkel, Daimler AG

Workshop der ASIM/GI-Fachgruppen Simulation technischer Systeme (STS) und Grundlagen und Methoden in Modellbildung und Simulation (GMMS), Wolfenbüttel 23./24.02.2012

Original equipment manufacturers (OEMs) and component suppliers have enormously increased the efficiency of electronic control unit (ECU) testing by using hardware-in-the-loop (HIL) simulation. The advantages are obvious: HIL tests are reproducible and reusable. With test automation, they can even run 24 hours a day, 7 days a week.

By means of HIL technology, function tests can be shifted to earlier development stages to increase the maturity of new software and/or electronics components. Without simulation-based test automation, expensive and time-consuming test drive cycles are performed directly in a vehicle or on conventional test benches. Test automation software provides broadly based access to the test system, e.g., to the real-time simulation model or to the electrical error simulation unit.

The goal of the ASAM HIL API standardization efforts is to allow for more reuse in test cases and to decouple test automation software from test hardware. This has been done by defining different access ports, e. g. Model Access (MAPort) and Electrical Error Simulation Access (EESPort), for a HIL system in the HIL API Project 1.0, released in 2009.

This presentation shows how test cases are decoupled from different test systems of different vendors. Furthermore, by means of the ASAM HIL API standard, test development can be shifted to earlier development stages thanks to the virtualization of expensive and limited HIL hardware. Thus, engineers can develop test cases in Virtual ECU Testing environments to initially check the consistency of the ECU software and its interfaces, task scheduling, etc.

Some ports are not HIL-specific. The MAPort, for example, can also be used to adapt simulation tools. This allows engineers to develop test cases in very early stages and in different domains in order to reuse them in later stages at a real HIL Simulator using HIL API. The Functional Mock-up Interfaces (FMI) initiative is a cooperation within the current HIL API Project 2.0. As a result of the ITEA2-funded project Modelisar, standardized interfaces for model exchange and cosimulation of subsystems from different domains have been developed. These “functional mock-up interfaces” will support simulation system setup at all stages of function software development (MIL, SIL, HIL, etc.). The plan is to release a subset of ASAM HIL API 2.0 mainly dealing with the MAPort and simulator control as “Functional Mock-up Interface for Applications”. This means that tests written in those early simulation environments can be directly reused in real HIL environments at a later stage.

Dr. Rainer Rasche, dSPACE GmbH  
Dr. Dietmar Neumerkel, Daimler AG

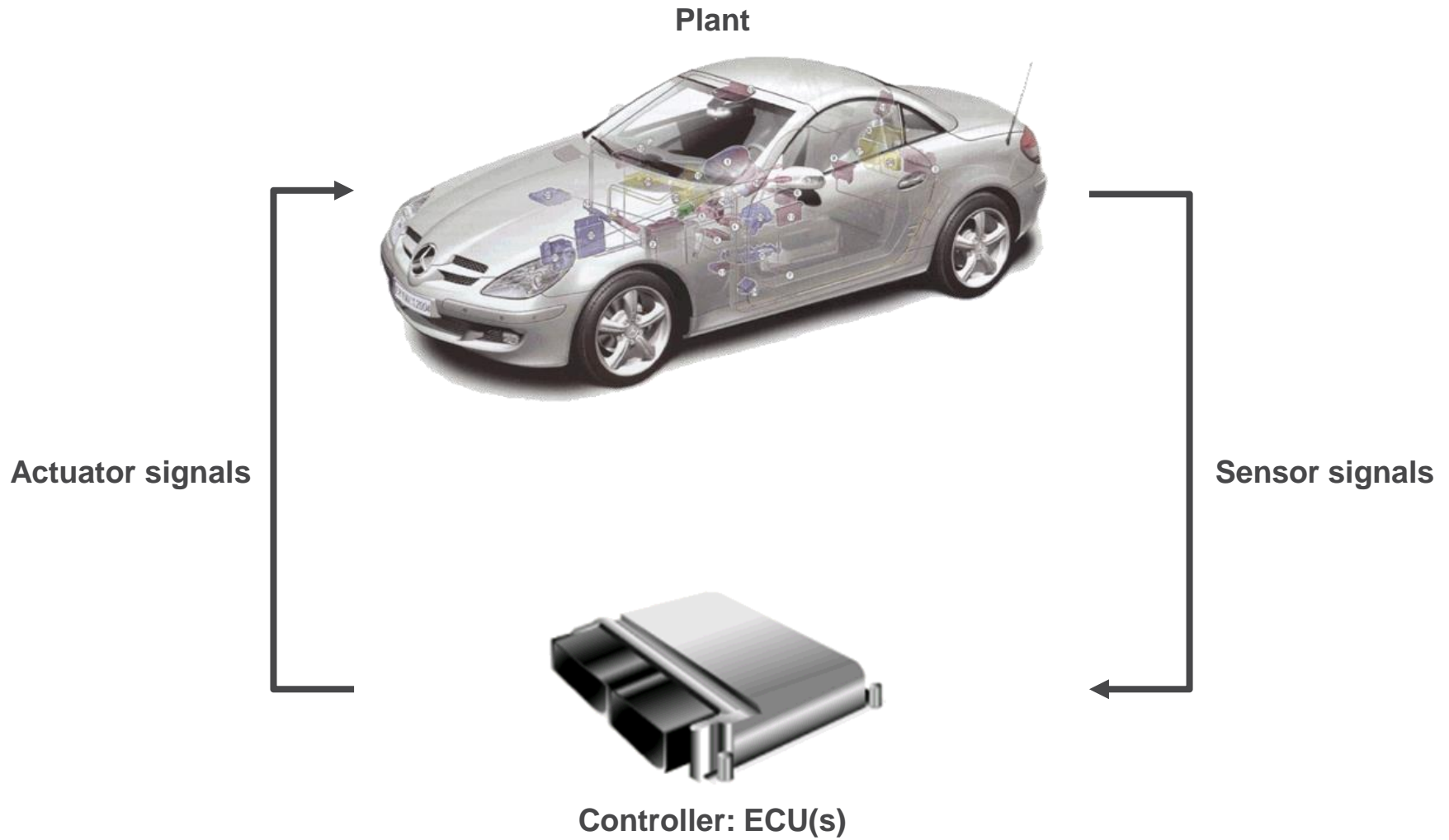
## Introduction: HIL Simulation

Testautomation ASAM HIL API 1.0

Improvements in ASAM HIL API 2.0

Example

Summary



Reality



Physical Principles

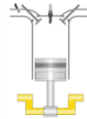


Simulation Model

Environment



Engine



Drivetrain



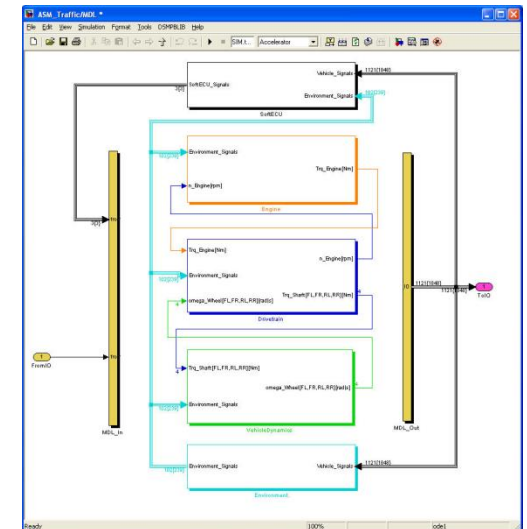
Vehicle Dynamics



ECU



ECU

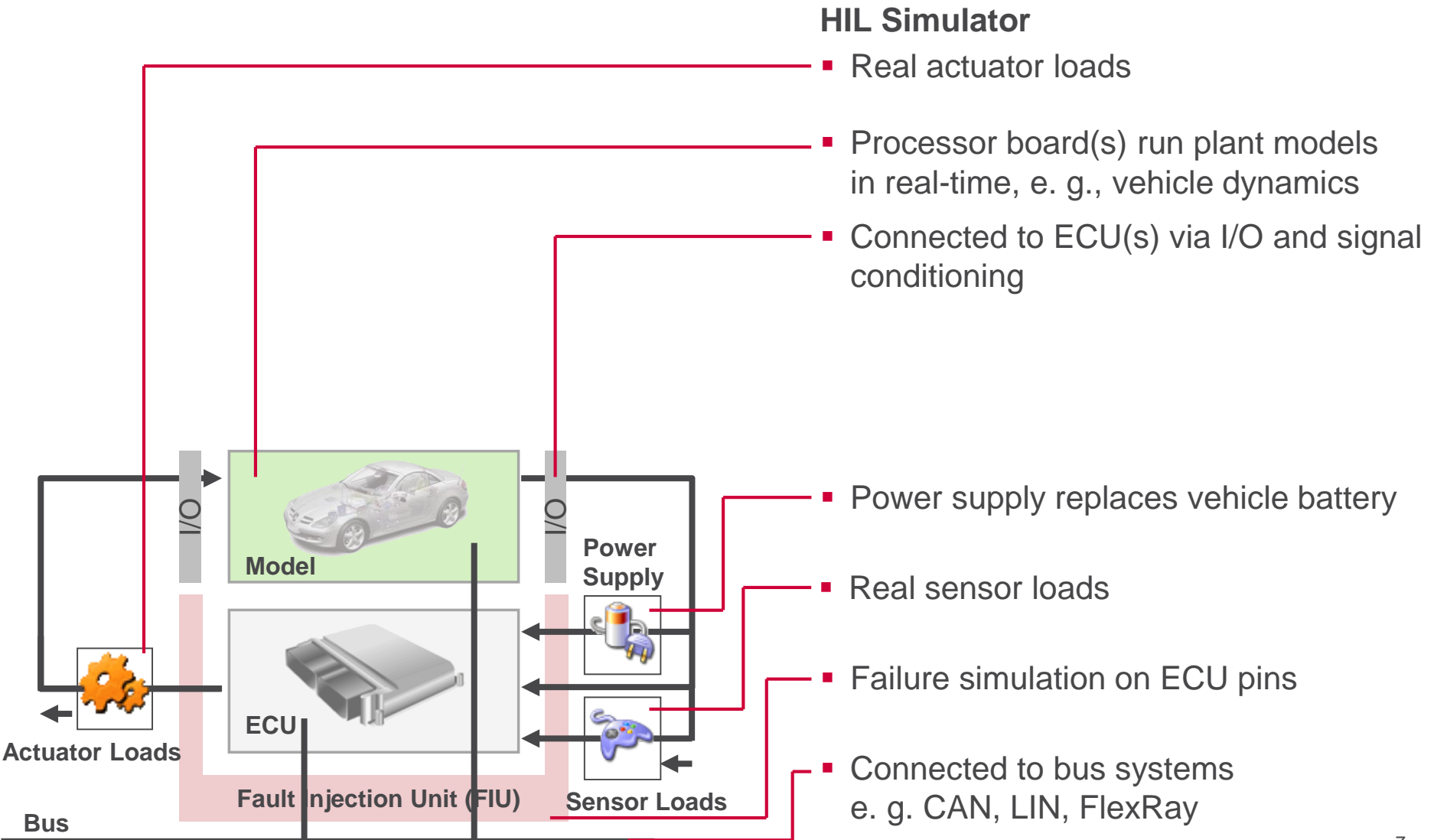


## Advantages

- Early testing without real vehicle prototypes
- Modifying test parameters easily
- Avoiding dangerous situations
- Avoiding abrasion, resource consumption
- Automated testing possible







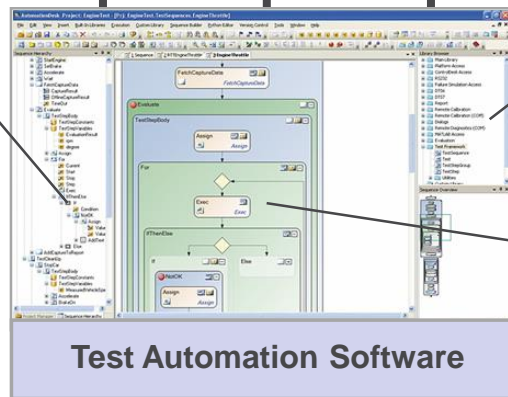
Requirements  
Engineering

Test  
Management

Version  
Control

## Test Project

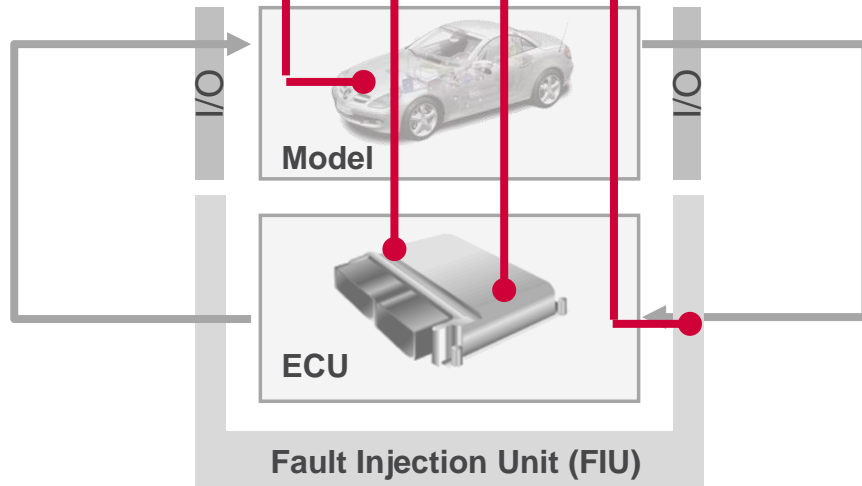
- Tests
- Data
- Results
- Reports



Test Library

Test Case

Test Automation Software



## Automated ECU Testing

- Repeating tests precisely and automatically as often as required
- Access to all relevant test interfaces
- State-of-the-art:  
Convenient PC-based  
test development and execution

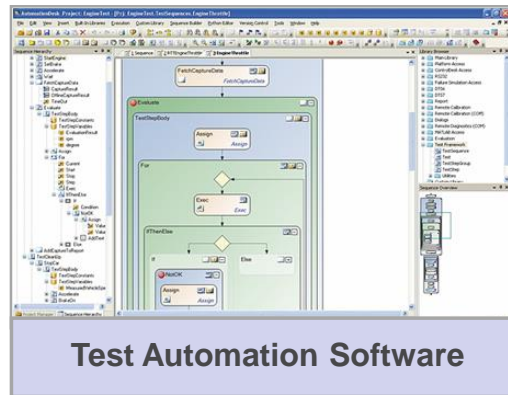
Interfaces to HIL test bench

Real-time simulation

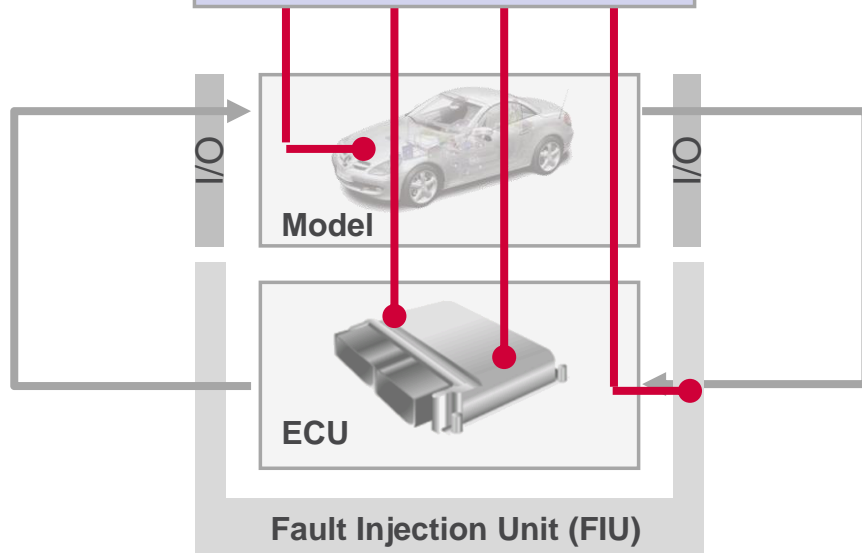
Reality



# Interfaces to HIL Test Bench: Today's Situation



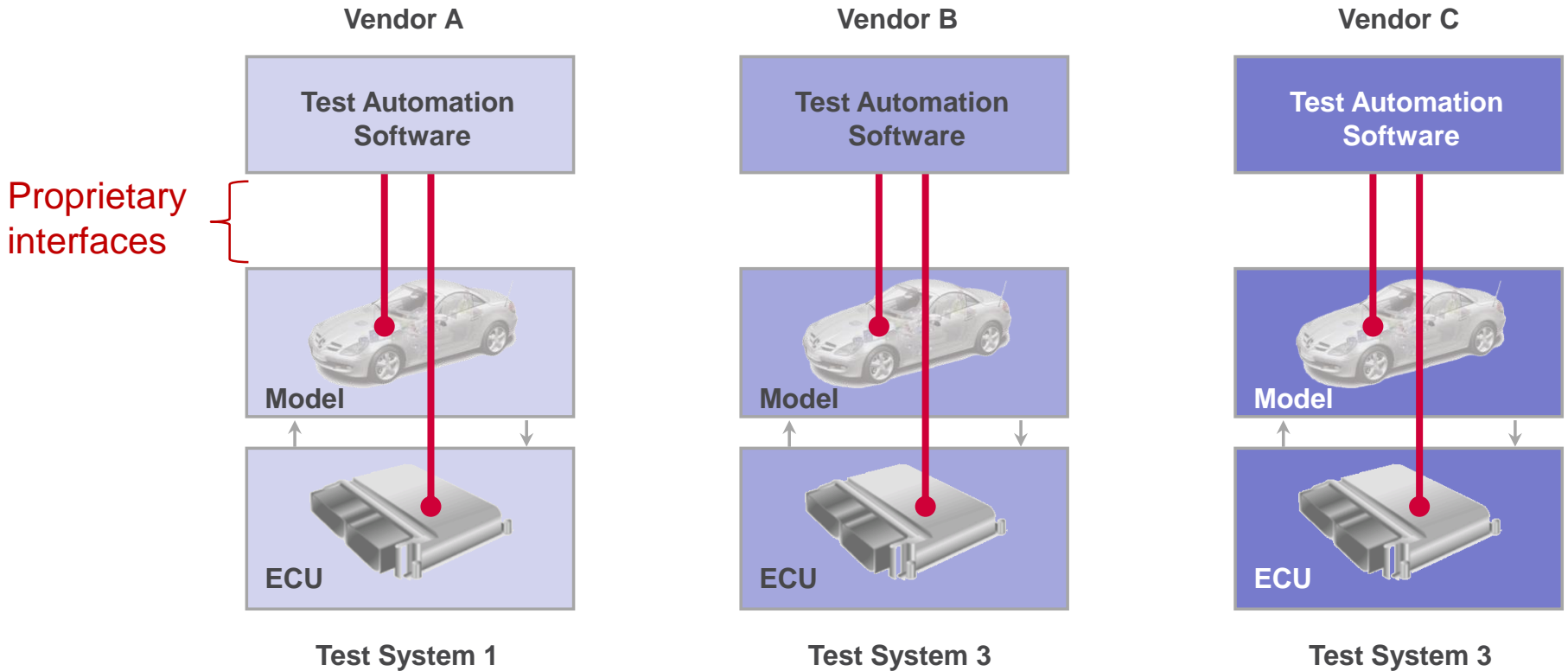
Test Automation Software



} Today, interfaces to HIL Test benches are often proprietary.

# Interfaces to HIL Test Bench: Today's Situation

Exchange of **test automation software** from different vendors and different **test systems** is **not** possible.



Introduction: HIL Simulation

Test Automation ASAM HIL API 1.0

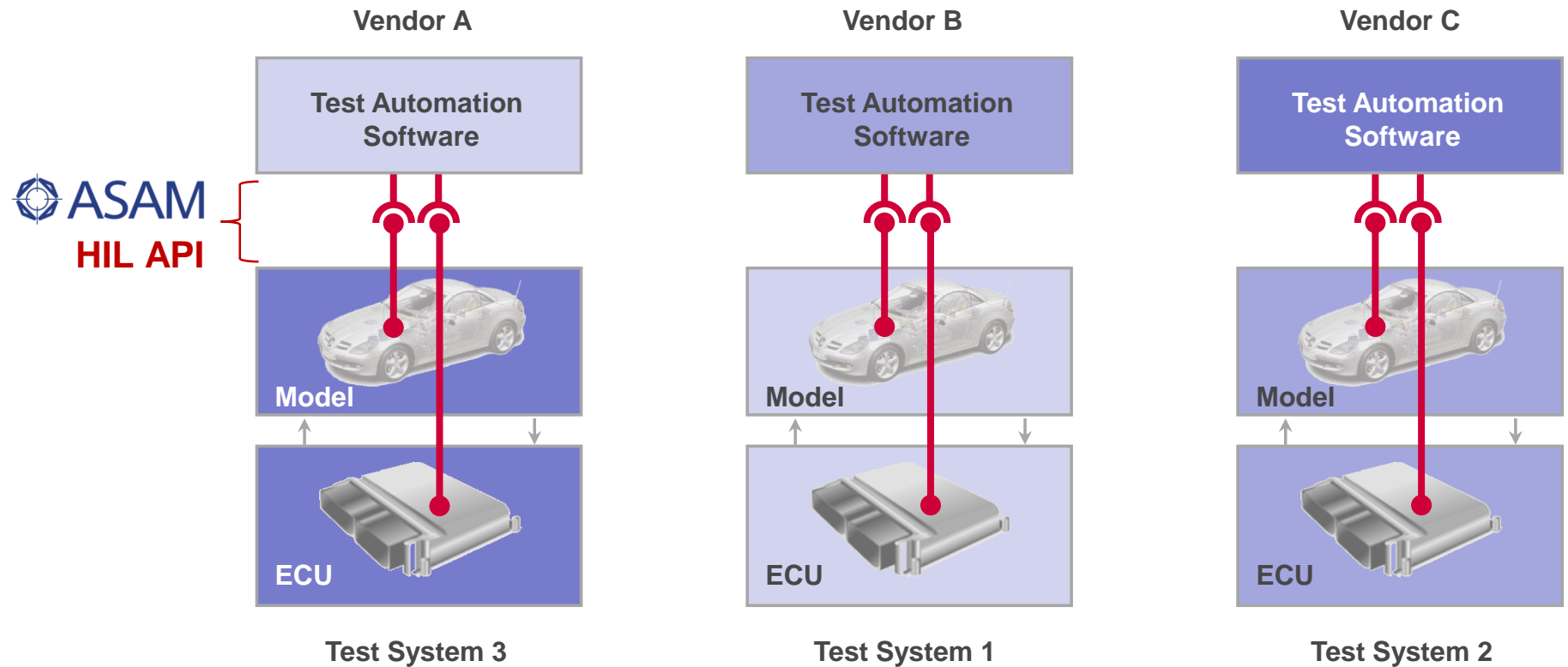
Improvements in ASAM HIL API 2.0

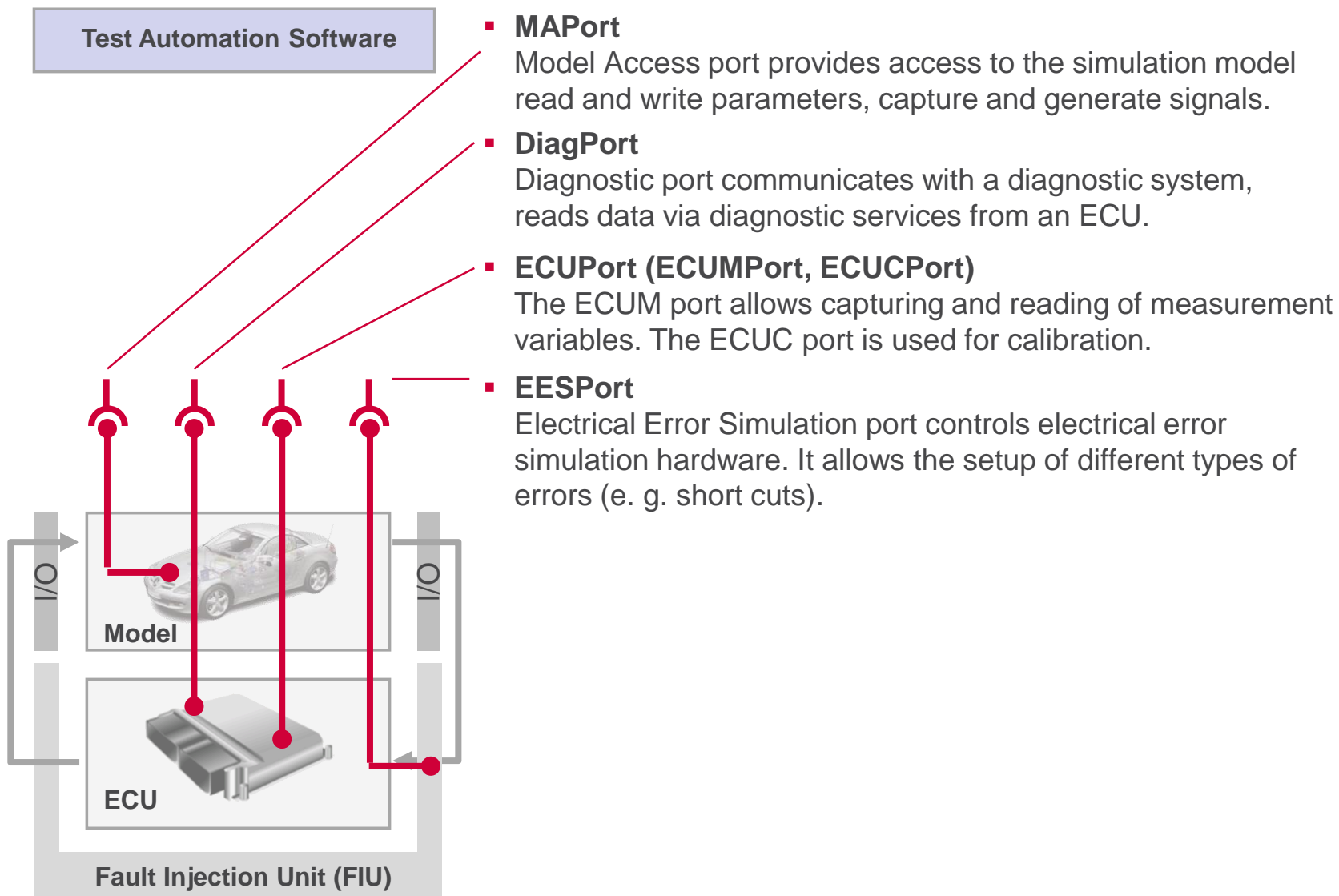
Example

Summary

# Interfaces to HIL Test Bench: Advantages of ASAM HIL API 1.0

Exchange of **test automation software** from different vendors and different **test systems** is possible.







## Real-time Simulation

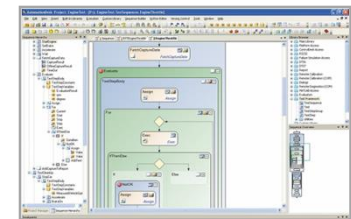
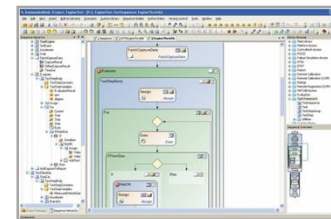
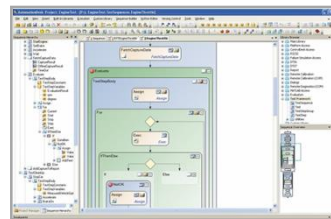
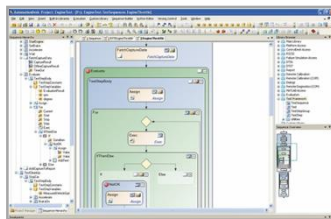
## Offline-Simulation

Real ECU

V-ECU

SiL or MiL

V-ECU

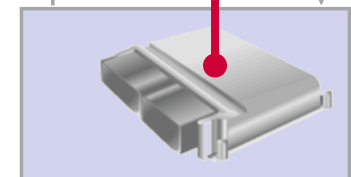
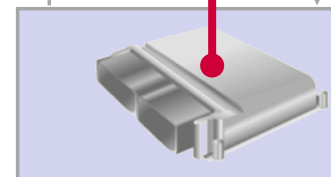
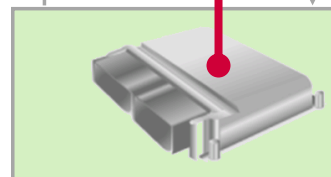
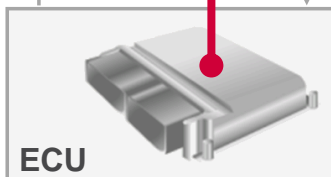
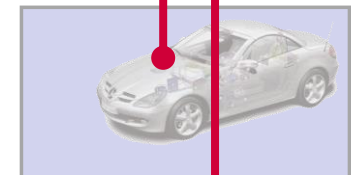
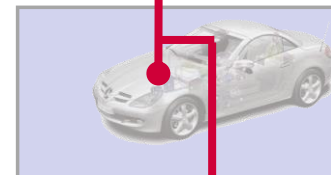
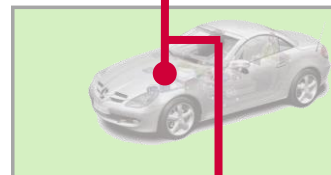
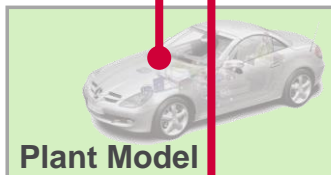


Test Automation  
Software

Test Automation  
Software

Test Automation  
Software

Test Automation  
Software



Reality

Real-time  
Simulation

PC-based  
Simulation



## Real-time Simulation

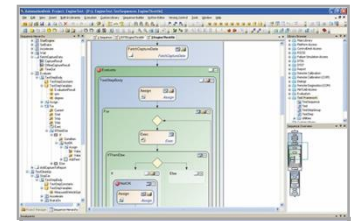
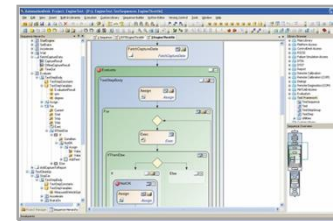
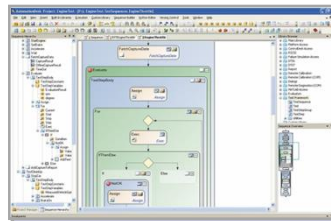
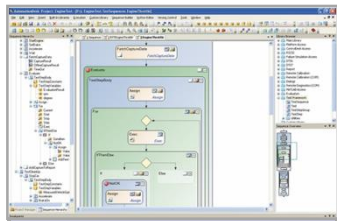
## Offline-Simulation

Real ECU

V-ECU

SiL or MiL

V-ECU



Test Automation  
Software

Test Automation  
Software

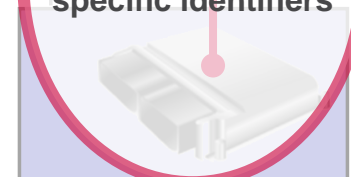
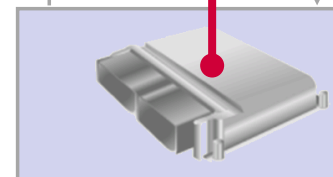
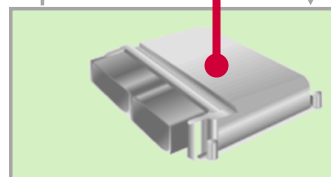
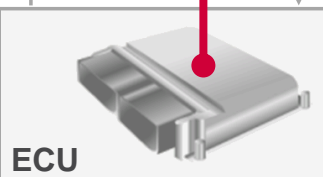
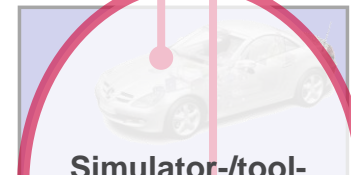
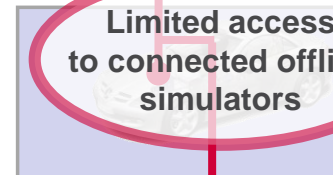
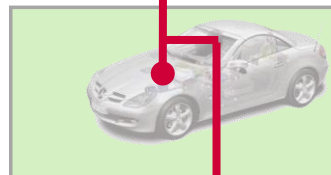
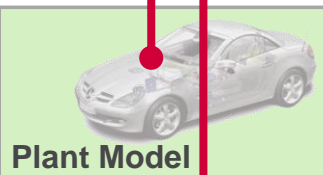
Test Automation  
Software

Test Automation  
Software

Port specific access

Limited access  
to connected offline  
simulators

Simulator-/tool-  
specific identifiers



ECU

Reality

Real-time  
Simulation

PC-based  
Simulation

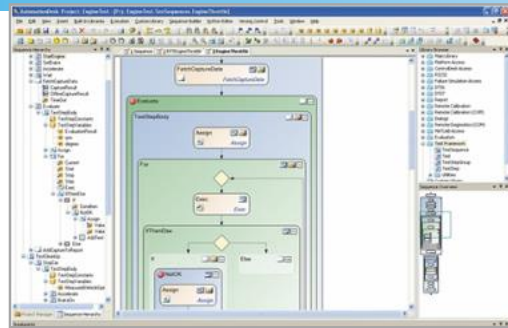
Introduction: HIL Simulation

Testautomation ASAM HIL API 1.0

Improvements in ASAM HIL API 2.0

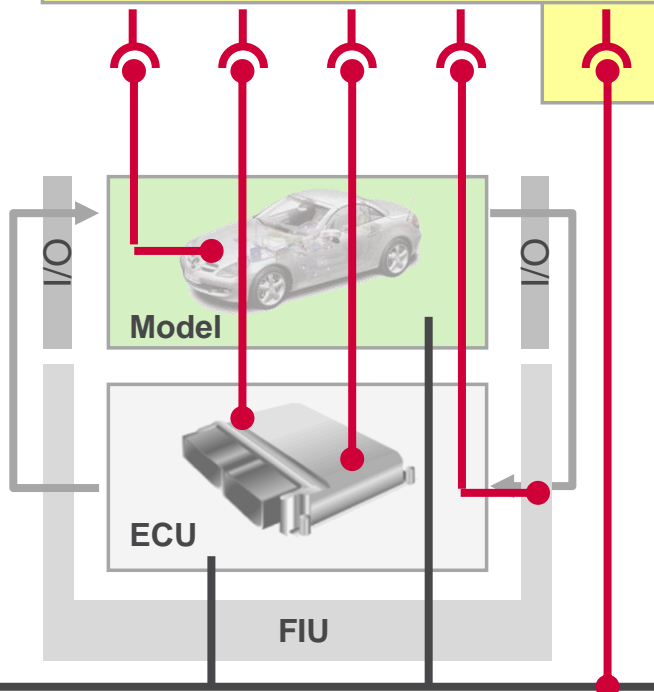
Example

Summary



Test Automation Software

HiL API 2.0 Framework



## Framework HIL API 2.0

- Mapping decouples test cases and HIL API 1.0 ports
- Simulator control allows standardized initialization of ports, start, stop simulation etc.

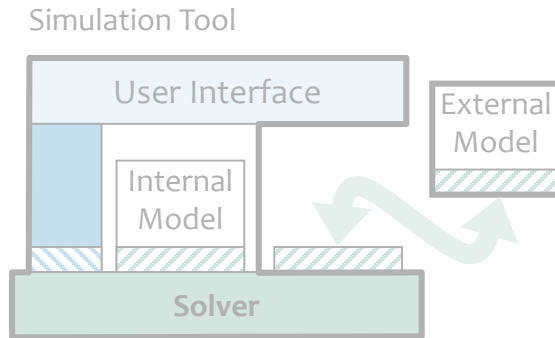
## Network Port:

Connects to the bus systems CAN, LIN, and FlexRay

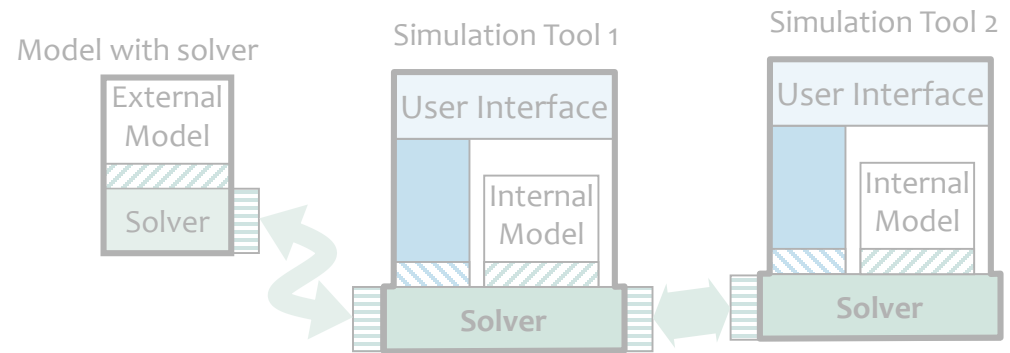
# Functional Mock-up Interfaces

## Access to Connected Offline Simulators

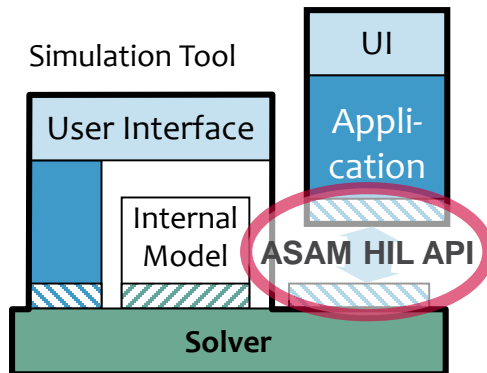
### 1. FMI for Model Exchange



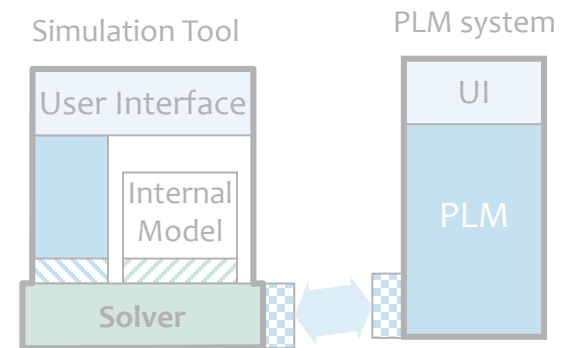
### 2. FMI for Co-Simulation



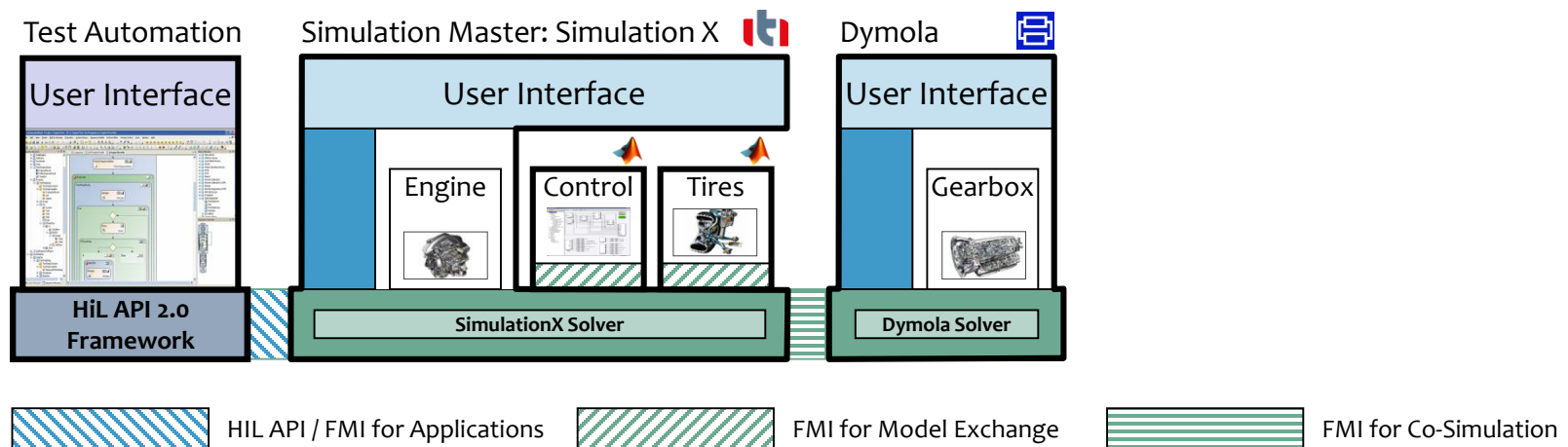
### 3. FMI for Applications



### 4. FMI for PLM (Product Lifecycle Management)



- Physical system is simulated by domain-specific tools
- Test automation is connected to master simulator via ASAM HiL API / FMI for Applications
- Master gives access to all subsystem parameters and signals
- Test automation starts / stops complete simulation system



Introduction: HIL Simulation

Testautomation ASAM HIL API 1.0

Improvements in ASAM HIL API 2.0

Example

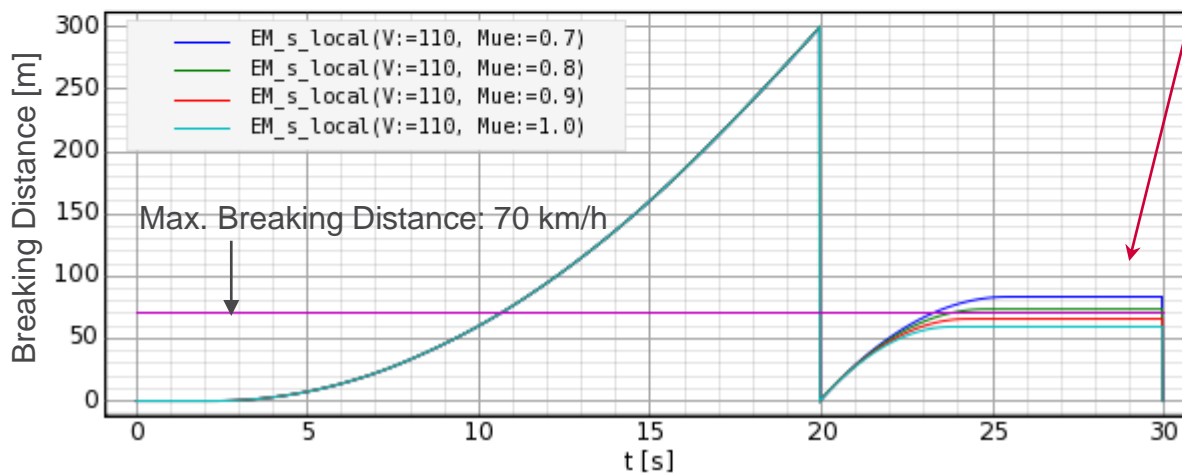
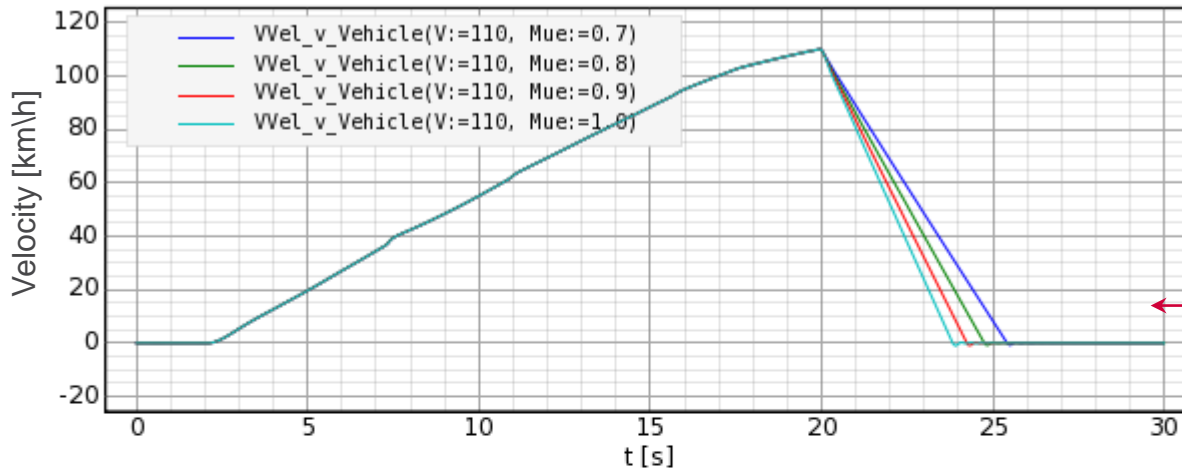
Summary

# Breaking Distance at Different Velocities and Mue Values

Test Results

Idx	Variant	Mue	Breaking Distance <= 70 m	Result
1	v80	0.7	45.4194598174	ok
2	v80	0.8	39.9757830826	ok
3	v80	0.9	35.6929258878	ok
4	v80	1.0	32.3389476788	ok
5	v90	0.7	55.9964128682	ok
6	v90	0.8	49.3076145999	ok
7	v90	0.9	44.0714312985	ok
8	v90	1.0	39.8734252759	ok
9	v100	0.7	70.3048877983	failed
10	v100	0.8	61.9330931503	ok
11	v100	0.9	55.3798247765	ok
12	v100	1.0	50.1041226752	ok
13	v110	0.7	83.6175271017	failed
14	v110	0.8	73.6794544687	failed
15	v110	0.9	65.9023631879	ok
16	v110	1.0	59.6391719084	ok

		Mue			
		0.7	0.8	0.9	1.0
V	80				
	90				
	100				
	110	83.6	73.7	65.9	59.6





# Breaking Distance at Different Velocities and Mue Values

		Mue			
		0.7	0.8	0.9	1.0
V	80				
	90				
	100				
	110	83.6	73.7	65.9	59.6

ABSBrakeTest v110 Mue: 0.8

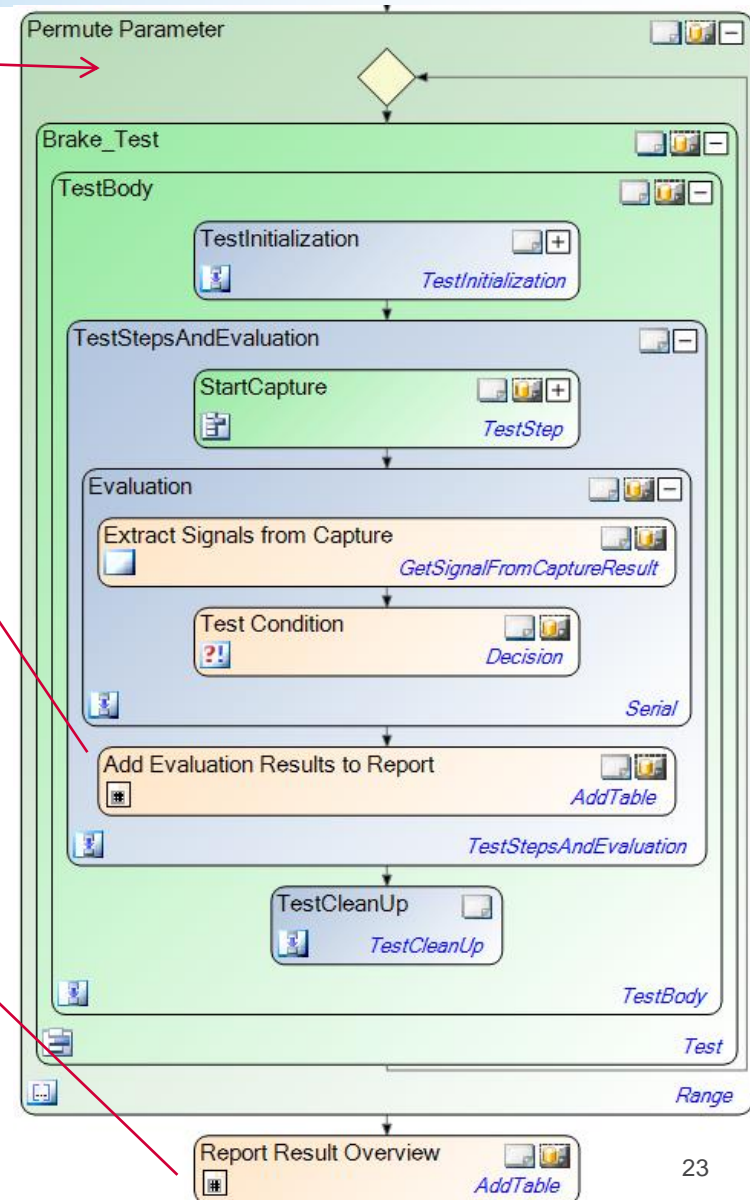
Result

EM\_CurrentSegment: 4 -- (19.943, 29.931)

Function	Value	Result	Comment
absMax(EM_s_local) <= 70	73.6794544687	failed	

## Test Results

Idx	Variant	Mue	absMax(EM_s_local) <= 70	Result
1	v80	0.7	45.4194598174	ok
2	v80	0.8	39.9757830826	ok
3	v80	0.9	35.6929258878	ok
4	v80	1.0	32.3389476788	ok
5	v90	0.7	55.9964128682	ok
6	v90	0.8	49.3076145999	ok
7	v90	0.9	44.0714312985	ok
8	v90	1.0	39.8734252759	ok
9	v100	0.7	70.3048877983	failed
10	v100	0.8	61.9330931503	ok
11	v100	0.9	55.3798247765	ok
12	v100	1.0	50.1041226752	ok
13	v110	0.7	83.6175271017	failed
14	v110	0.8	73.6794544687	failed
15	v110	0.9	65.9023631879	ok
16	v110	1.0	59.6391719084	ok



Introduction: HIL Simulation

Testautomation ASAM HIL API 1.0

Improvements in ASAM HIL API 2.0

Example

Summary

- **Today, test cases often directly depend on the used test hardware**
- **ASAM HIL API 1.0**  
Decouples test cases from real and virtual test systems using ports
- **ASAM HIL API 2.0**  
Mapping decouples test cases and HIL API 1.0 ports  
Standardized simulator control to initialize ports  
Network port supports CAN, LIN, FlexRay
- Easy test case exchange between coupled offline simulators in early stages and productive HIL test benches
- Better know-how transfer from one test bench to the other
- Reduced training costs for employees
- **From end users perspective:**  
**This allows the ‘best’ test software combined with the ‘best’ test hardware.**

# Thank you very much for your attention.

**dSPACE**



© Copyright 2012, dSPACE GmbH. All rights reserved.

Brand names or product names are trademarks or registered trademarks of their respective companies or organizations.

Contact: [rrasche@dspace.de](mailto:rrasche@dspace.de), [dietmar.neumerkel@daimler.com](mailto:dietmar.neumerkel@daimler.com)

Info: <http://www.asam.net>, <http://functional-mockup-interface.org>