

ASAM XIL 2.0: Standard für ein durchgängiges Testen im gesamten Produktentstehungsprozess (MiL/SiL/HiL)

7. Vector Congress, November 26, 2014, Stuttgart

Dr. Rainer Rasche, dSPACE GmbH (Speaker)

Dr. Dietmar Neumerkel, Daimler AG

Constantin Brückner, AUDI AG

Agenda

- | | |
|---|--------------------------------------|
| 1 | History |
| 2 | Introduction and Motivation |
| 3 | Concept of HIL API in Version 1.0 |
| 4 | Crosstests |
| 5 | Big Points of XIL API in Version 2.0 |
| 6 | Summary and Conclusion |

Agenda

- | | |
|---|--------------------------------------|
| 1 | History |
| 2 | Introduction and Motivation |
| 3 | Concept of HIL API in Version 1.0 |
| 4 | Crosstests |
| 5 | Big Points of XIL API in Version 2.0 |
| 6 | Summary and Conclusion |

History

- 2009 (July)
- 2010 (December):
- 2012 (January)
- 2012 (June)
- 2013 (February)
- 2013 (October)
- 2014 (October)
- Currently in progress

- HIL API 1.0.0
- HIL API 1.0.1
- HIL API 1.0.2
- Crosstest No. 1 (among 4 vendors)
- Crosstest No. 2 (among 5 vendors)
- XIL API 2.0.0
- XIL API 2.0.1
- XIL API 2.0.2

Agenda

1 History

2 Introduction and Motivation

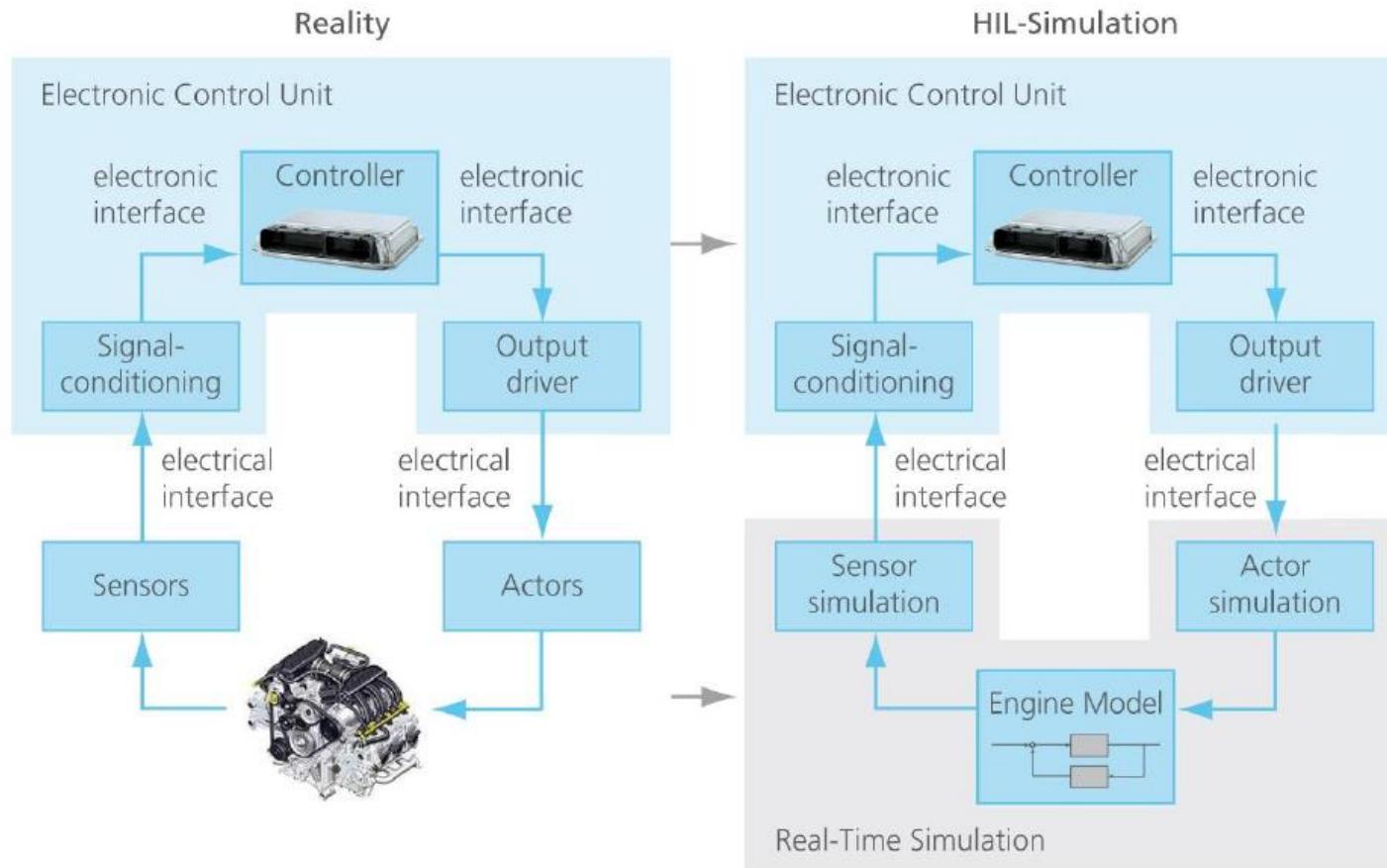
3 Concept of HIL API in Version 1.0

4 Crosstests

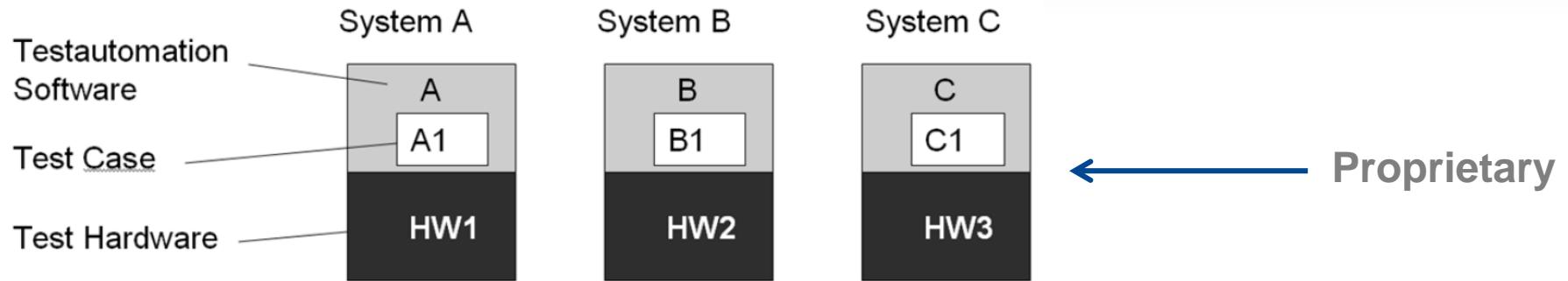
5 Extensions of XIL API in Version 2.0

6 Summary and Conclusion

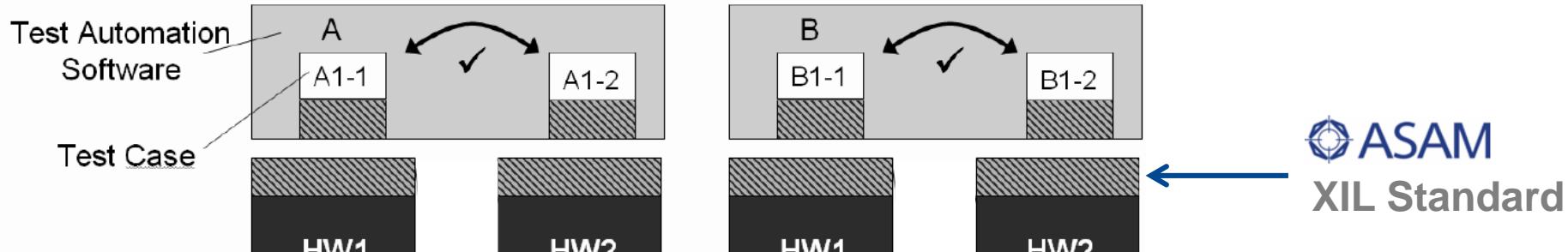
Principle of Hardware-in-the-Loop Simulation



Motivation of the Standard



→ Separation of Test HW and Test SW by means of standardized APIs



Agenda

1 History

2 Introduction and Motivation

3 Concept of HIL API in Version 1.0

4 Crosstests

5 Extensions of XIL API in Version 2.0

6 Summary and Conclusion

Concept of Ports

MAPort

Model Access port provides access to the simulation model read and write parameters, capture and generate signals.

NetworkPort

provides access to field bus systems such as CAN. E. g. Allows measurement (monitoring) and transmission (single transmit or replay) of bus data.

EESPort

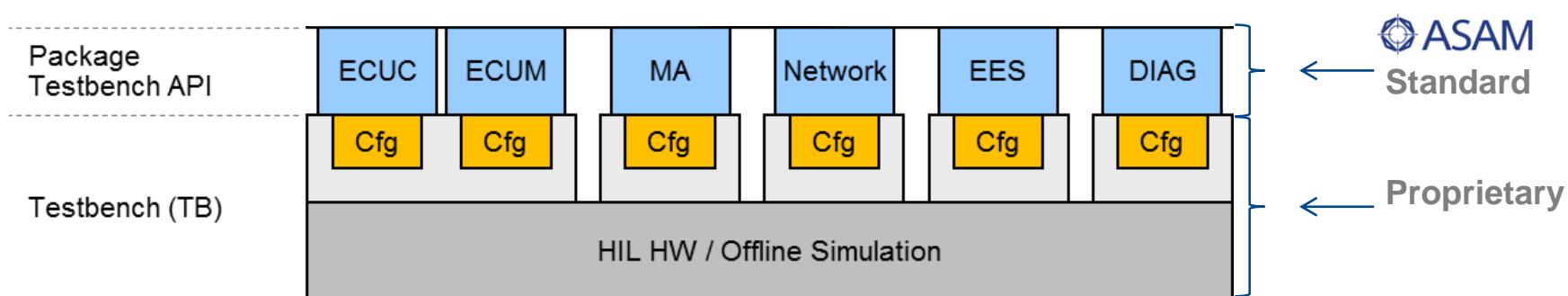
Electrical Error Simulation port controls electrical error simulation hardware. It allows the setup of different types of errors (e. g. short cuts).

ECUPort

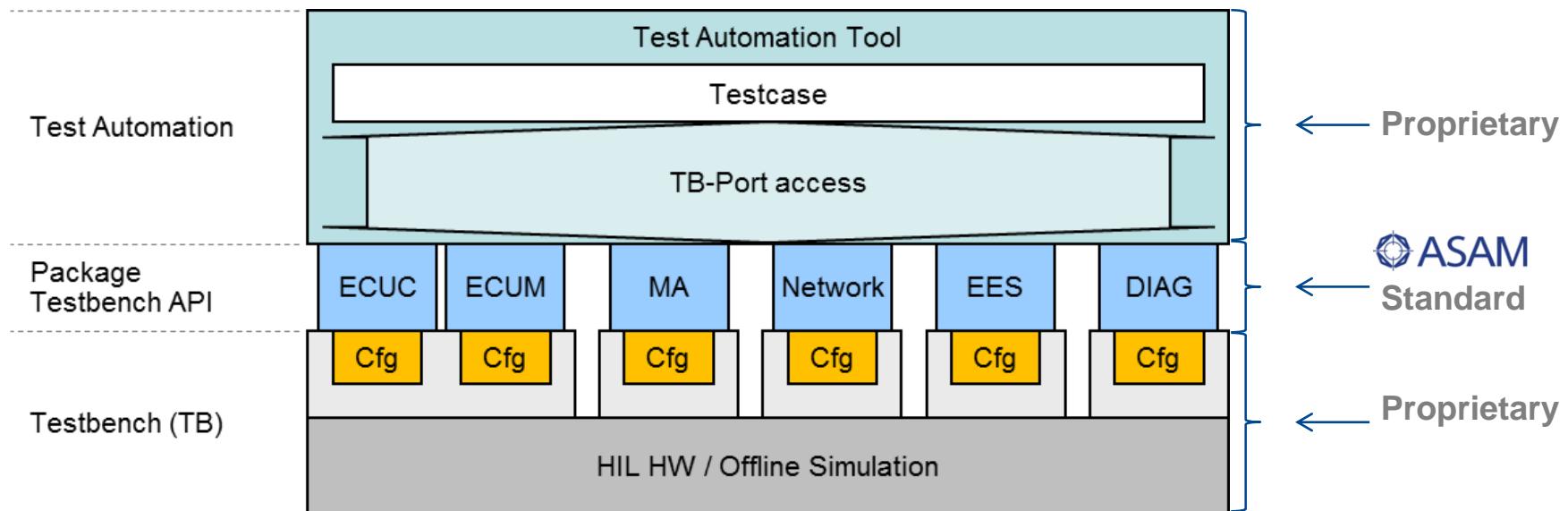
The ECUM port allows capturing and reading of measurement variables. The ECUC port is used for **calibration**.

DiagPort

Diagnostic port communicates with a diagnostic system, reads data via diagnostic services from an ECU.



Port-based Access of Test Automation Tools

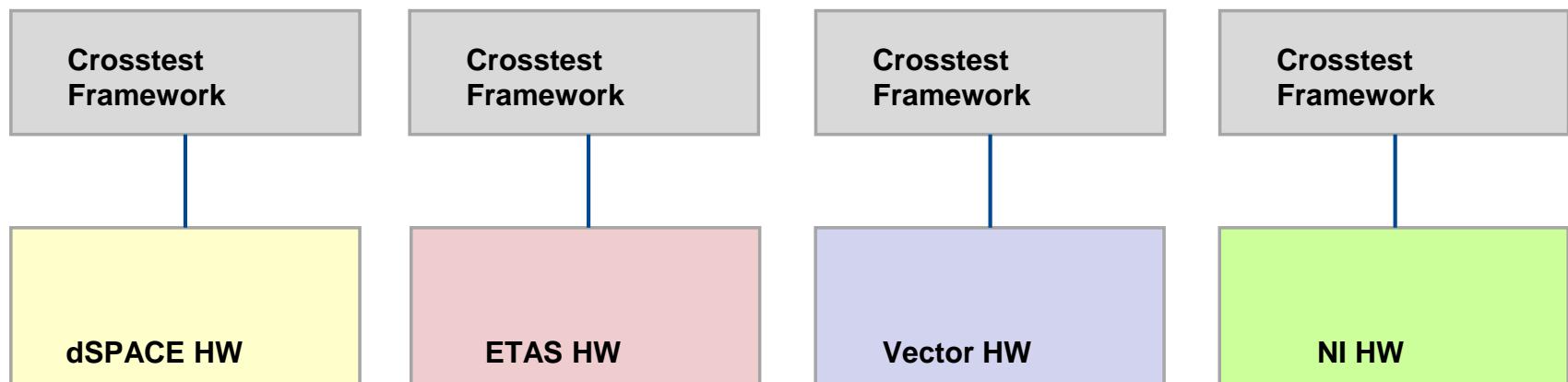


Agenda

- | | |
|---|--------------------------------------|
| 1 | History |
| 2 | Introduction and Motivation |
| 3 | Concept of HIL API in Version 1.0 |
| 4 | Crosstests |
| 5 | Extensions of XIL API in Version 2.0 |
| 6 | Summary and Conclusion |

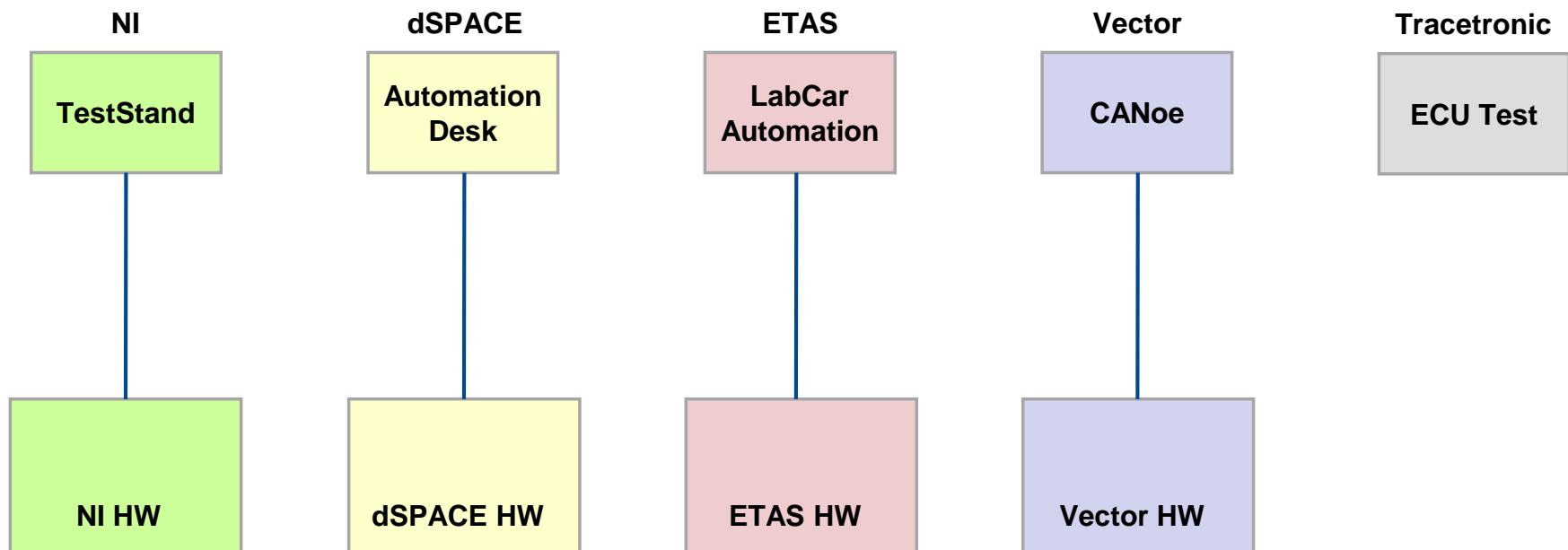
Cross Test (Stage 1)

- ▶ Test components:
 - 49 pre-defined test cases implemented in C# (XIL Standard)
 - Matlab/Simulink model of a throttle valve controller
 - C# Test Framework (“lightweight” client)
- ▶ All test components were provided before the cross test
- ▶ Each participant were asked to implement the vendor specific part of the test Framework and check if the pre-defined test cases could be executed successfully



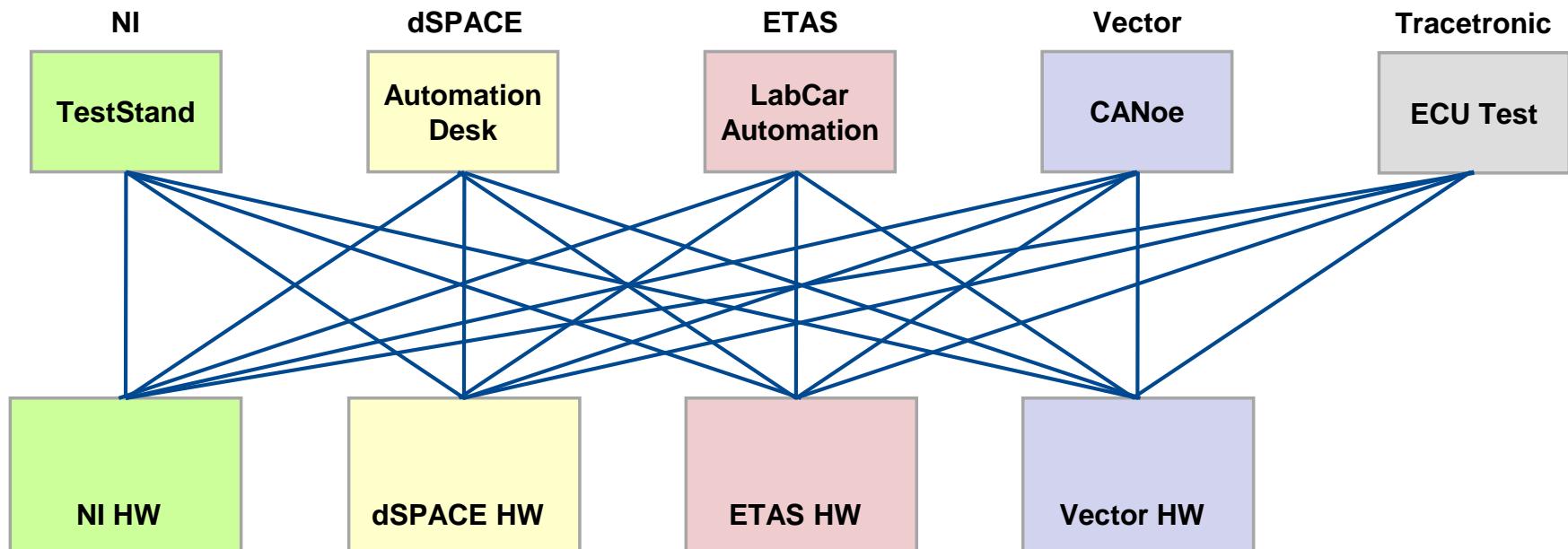
Cross Test (Stage 2)

- ▶ Pre-defined test cases of stage 1 are implemented and executed using the “real-world” clients of each vendor
- ▶ 20 possible combinations
- ▶ It is checked if the tests can be executed and test results are compared



Cross Test (Stage 2)

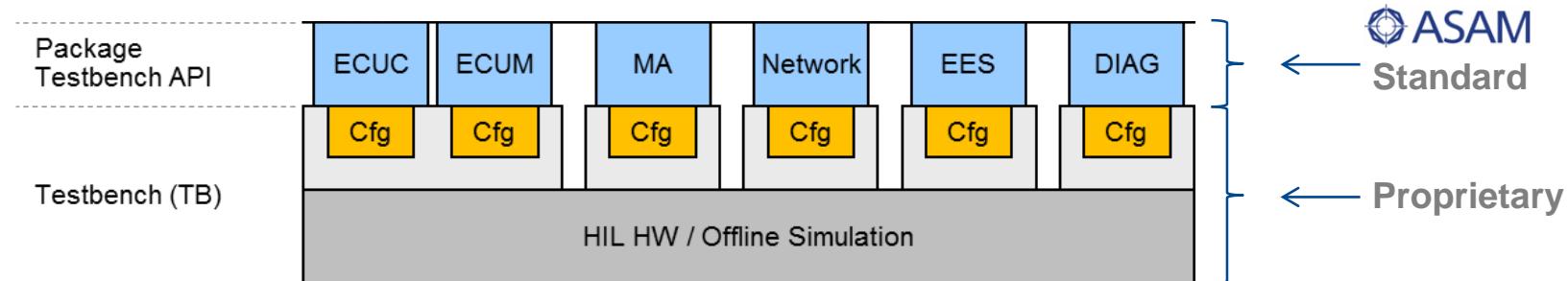
- ▶ Pre-defined test cases of stage 1 are implemented and executed using the “real-world” clients of each vendor
- ▶ 20 possible combinations
- ▶ It is checked if the tests can be executed and test results are compared



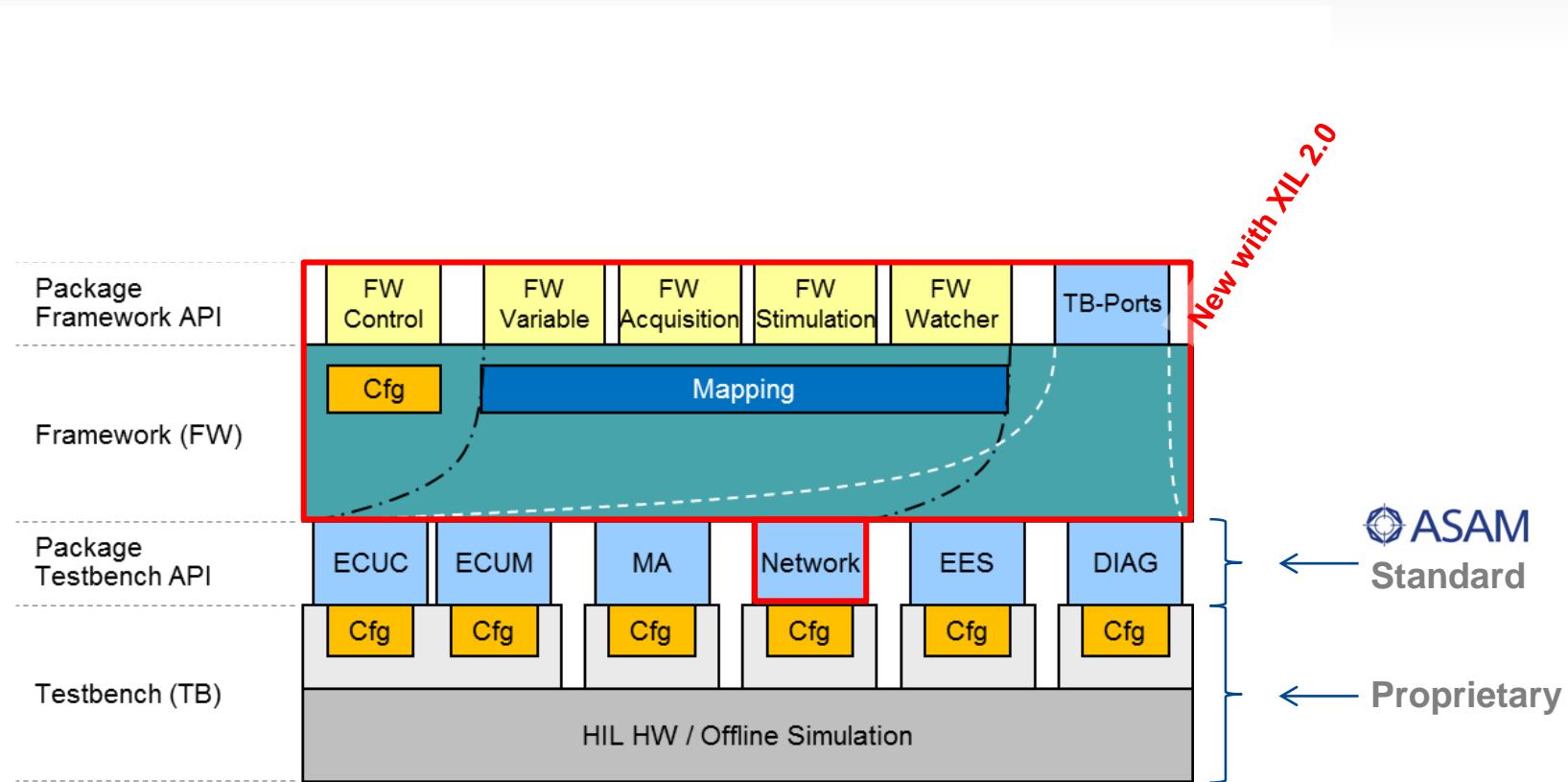
Agenda

- | | |
|---|--------------------------------------|
| 1 | History |
| 2 | Introduction and Motivation |
| 3 | Concept of HIL API in Version 1.0 |
| 4 | Crosstests |
| 5 | Extensions of XIL API in Version 2.0 |
| 6 | Summary and Conclusion |

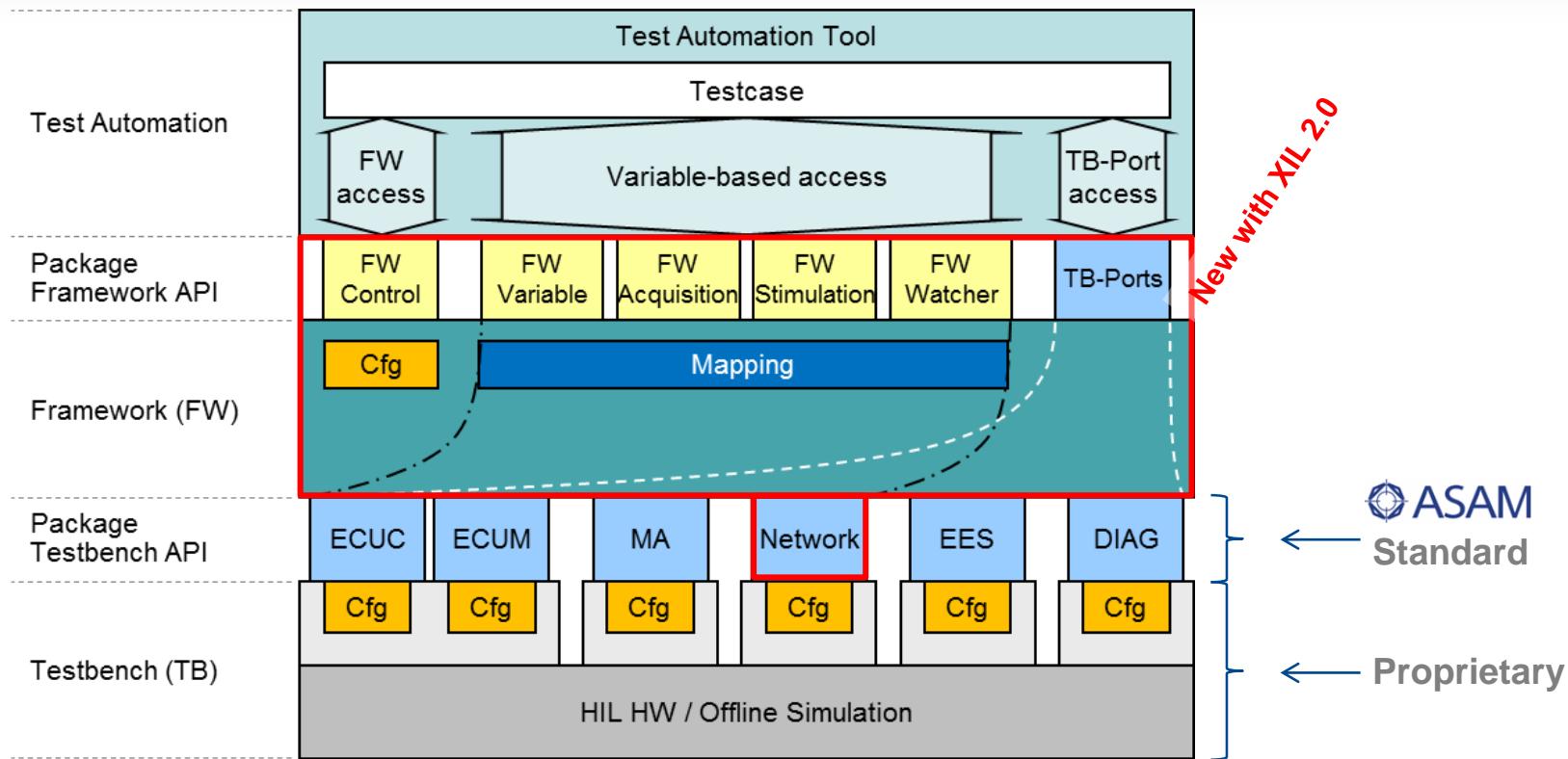
Framework-based Access with XIL 2.0



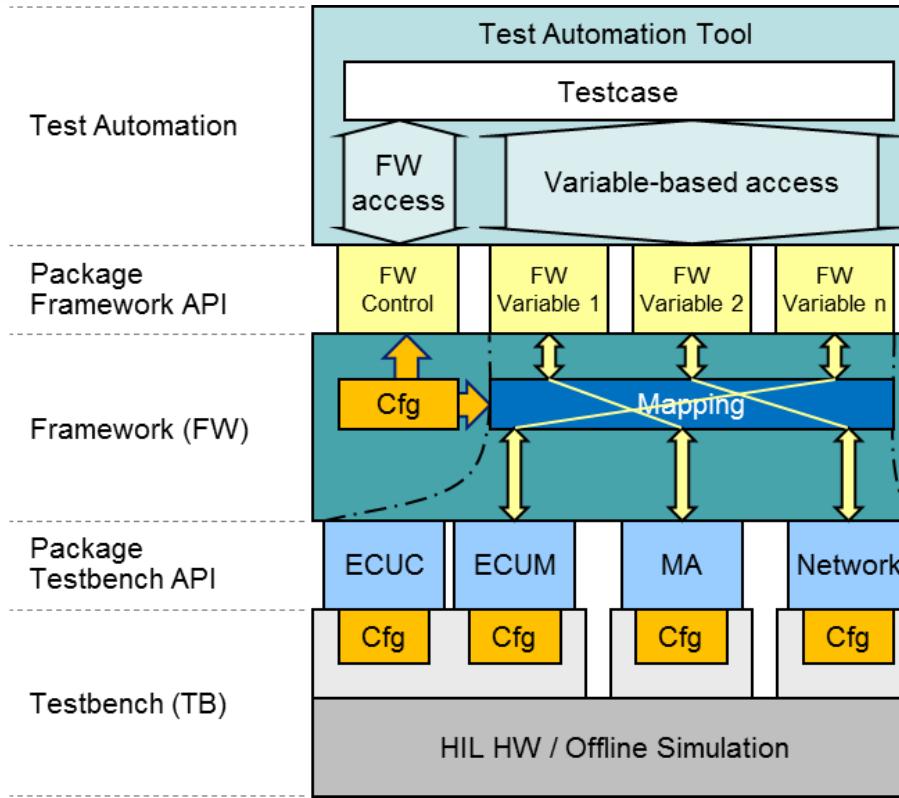
Framework-based Access with XIL 2.0



Framework-based Access with XIL 2.0



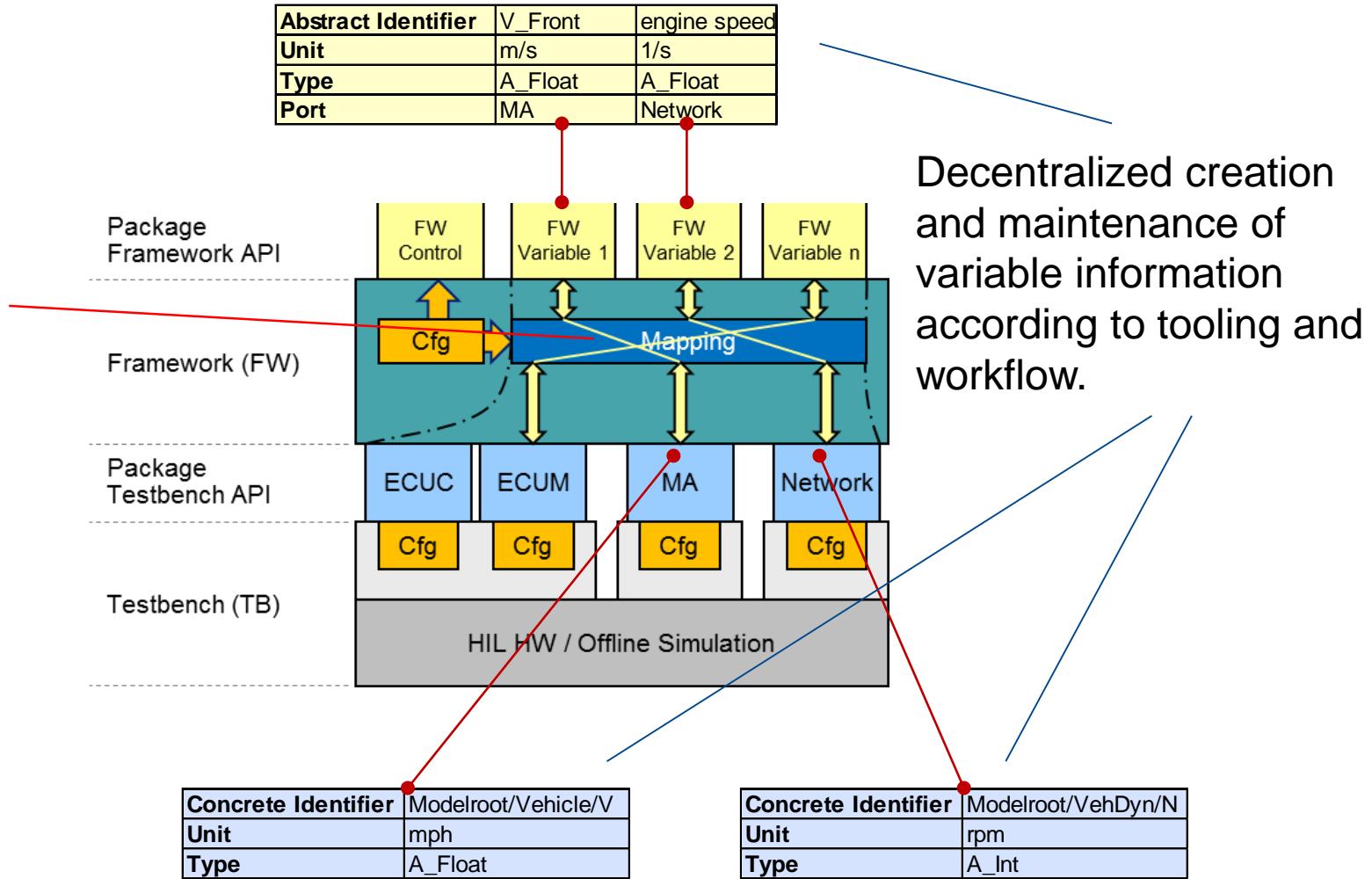
Framework Variable



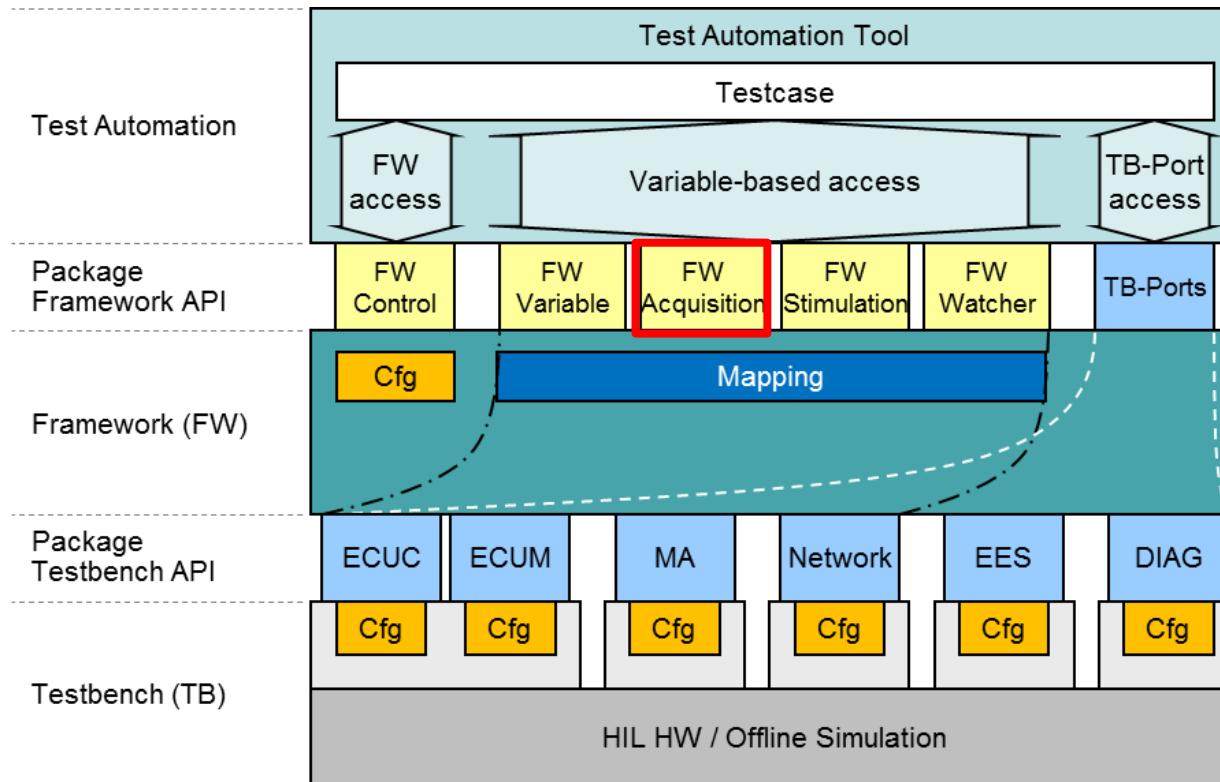
- ▶ Abstract Identifiers on testcase side (e. g. “engine speed”).
- ▶ Guarantees that testcases are independent of the underlying test system
- ▶ **Thus, independent of vendor and process stage → XIL**

Mapping

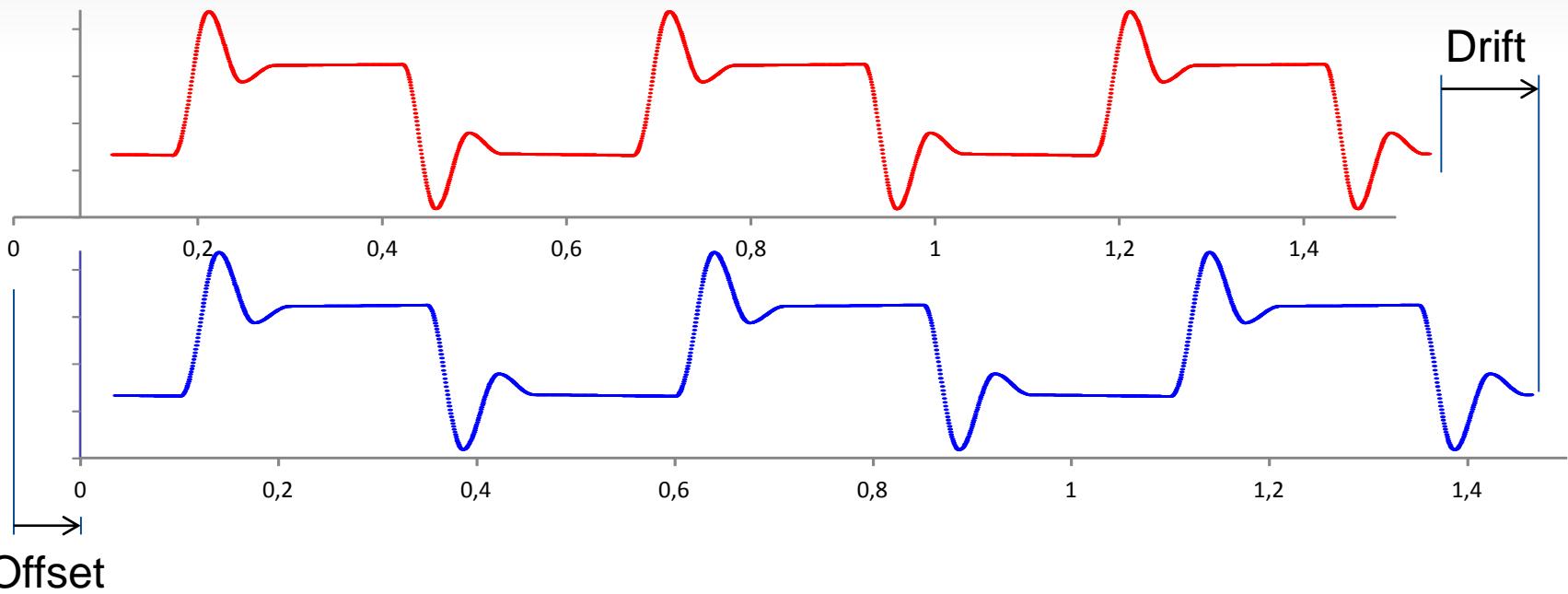
Centralized mapping can import decentralized information.



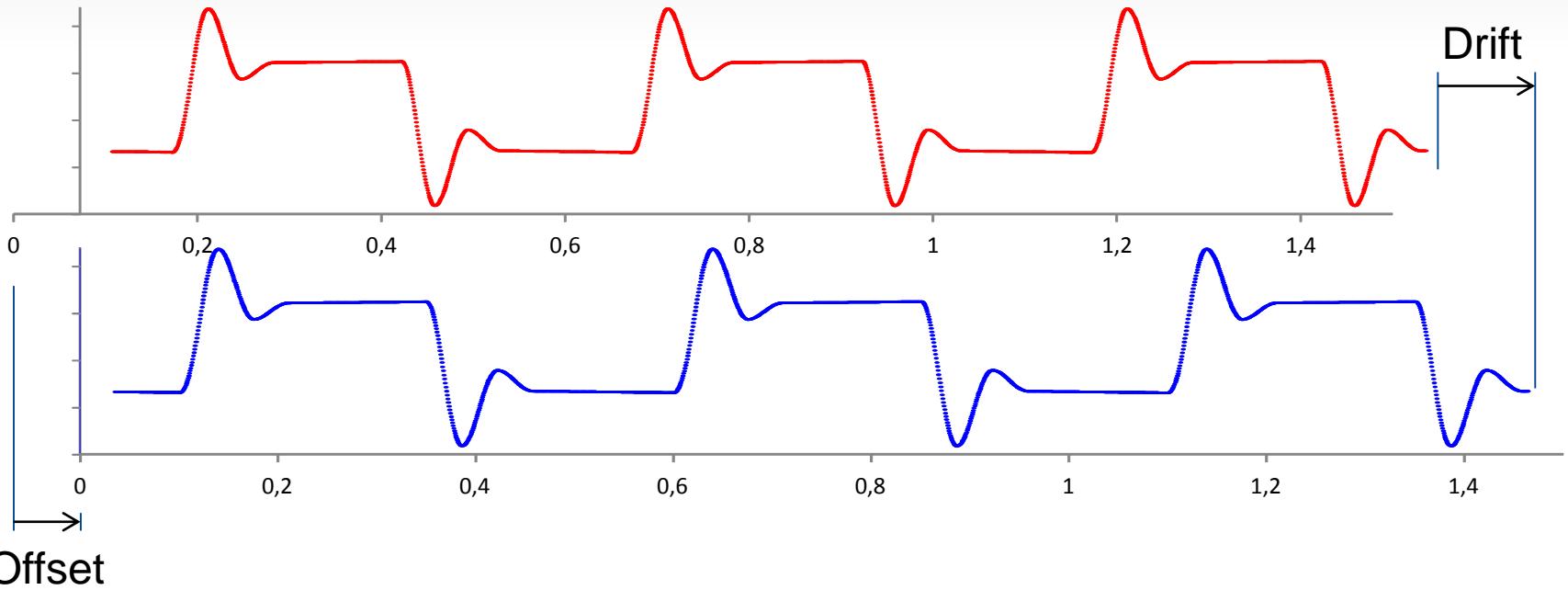
Data Acquisition



Synchronization

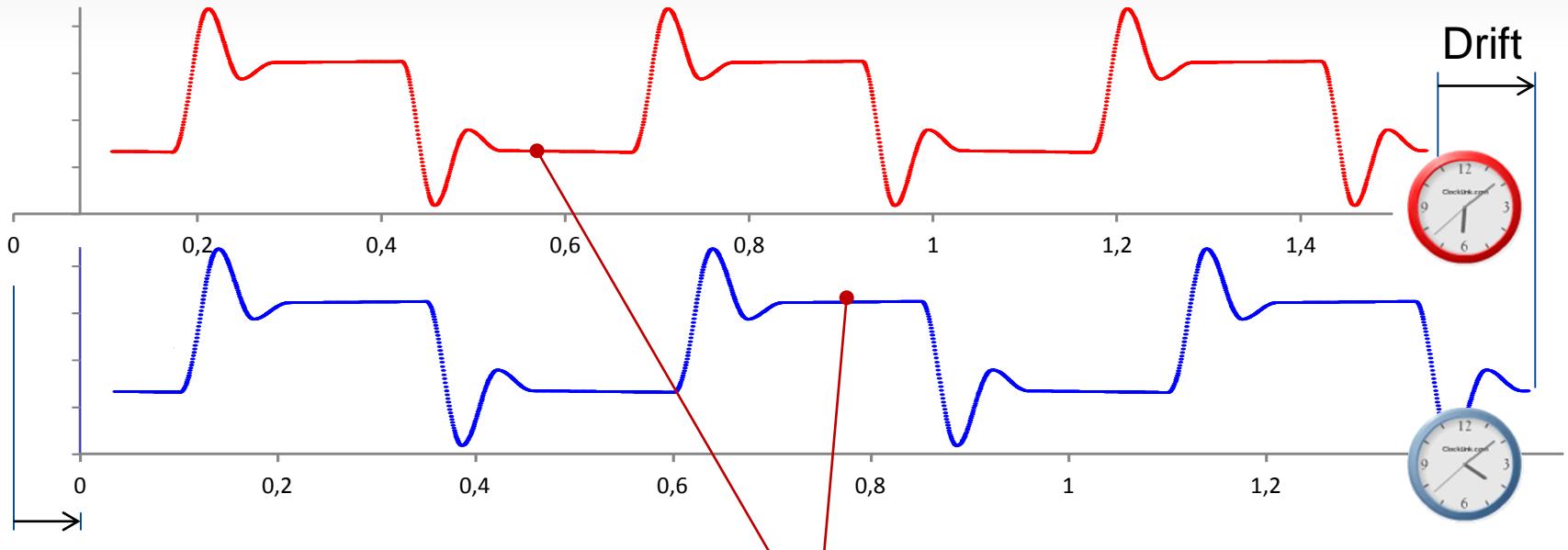


Synchronization

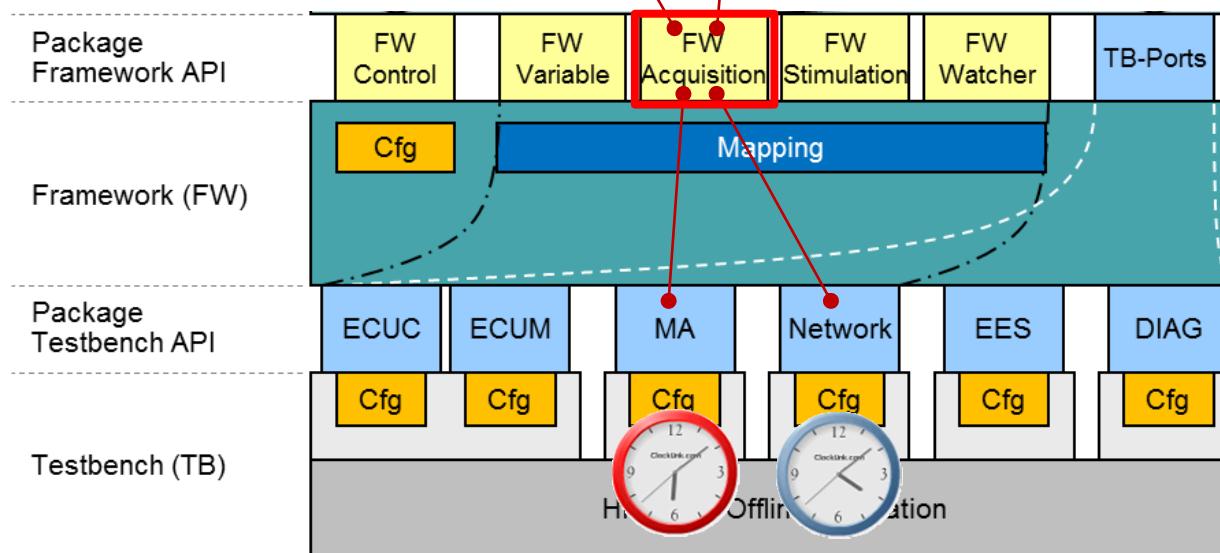


- ▶ Different hardware times of different signals sources usually have
 - Offset
 - Drift
- ▶ XIL 2.0.0 provides interfaces to configure compensation of offset and drift on a common time base

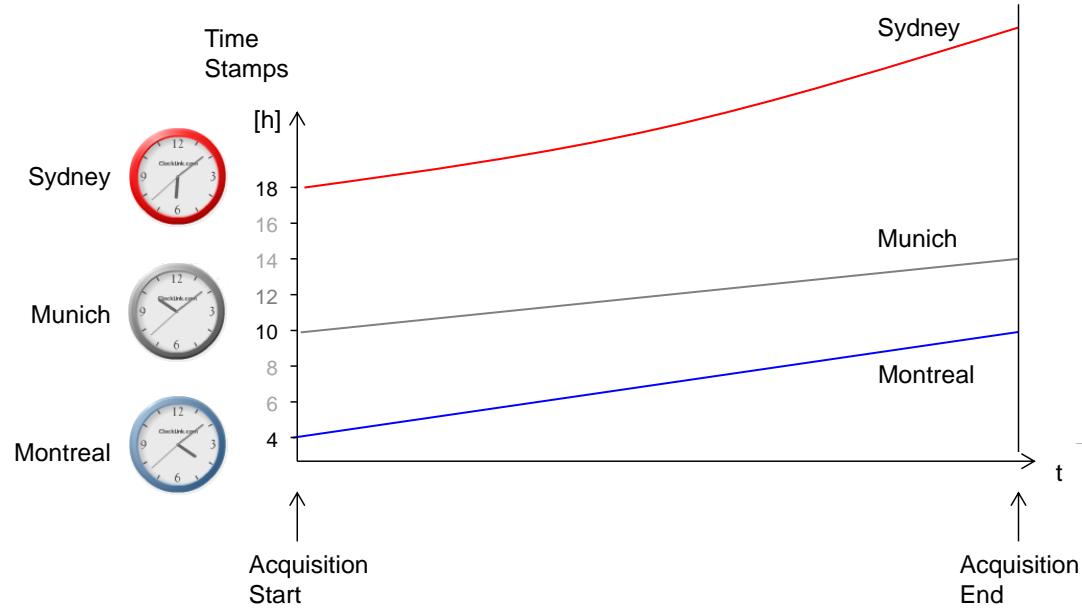
Synchronization



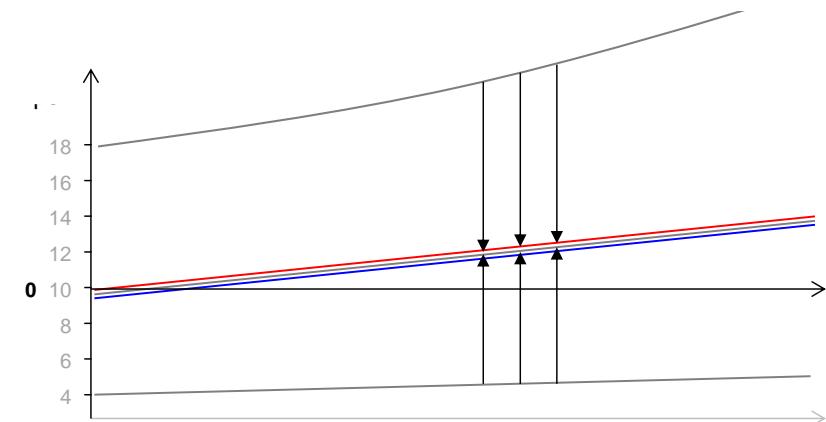
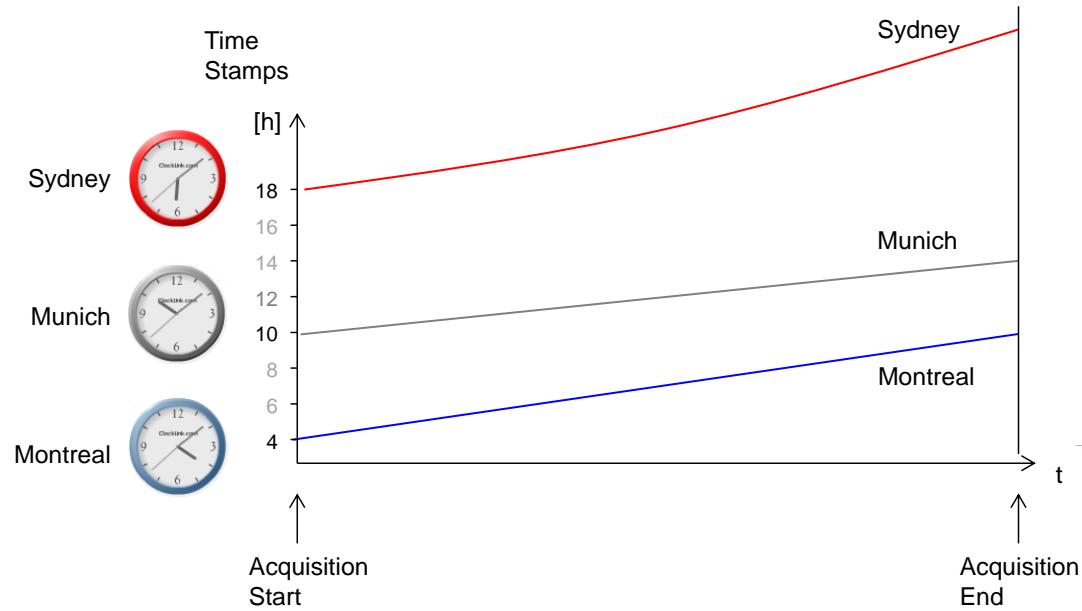
Offset



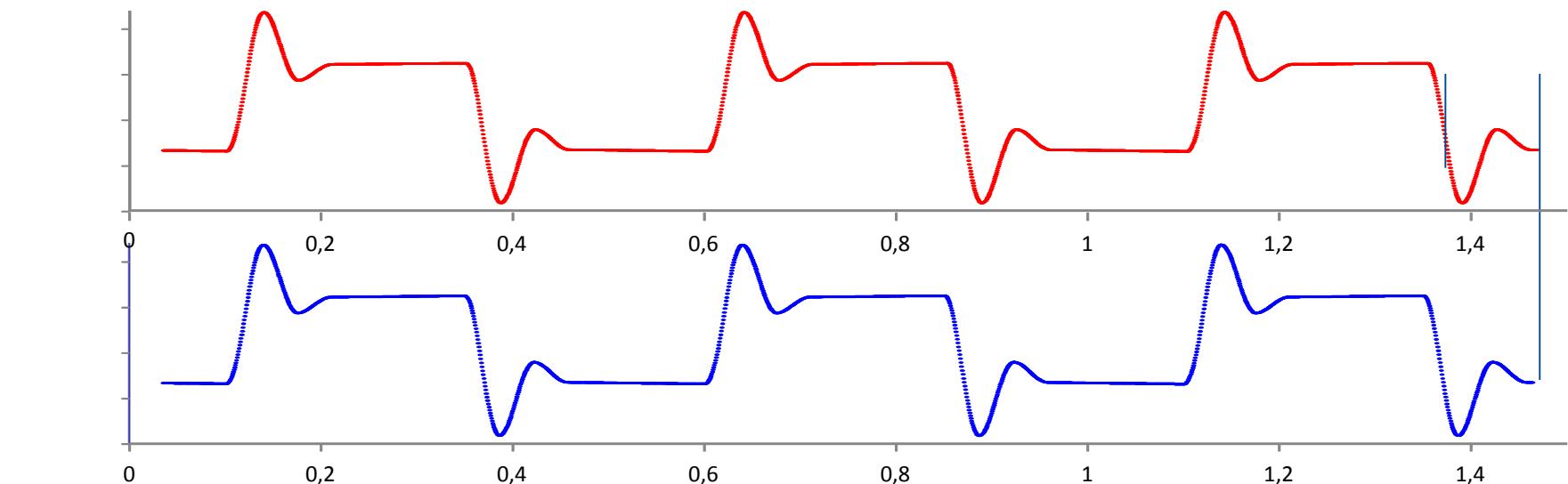
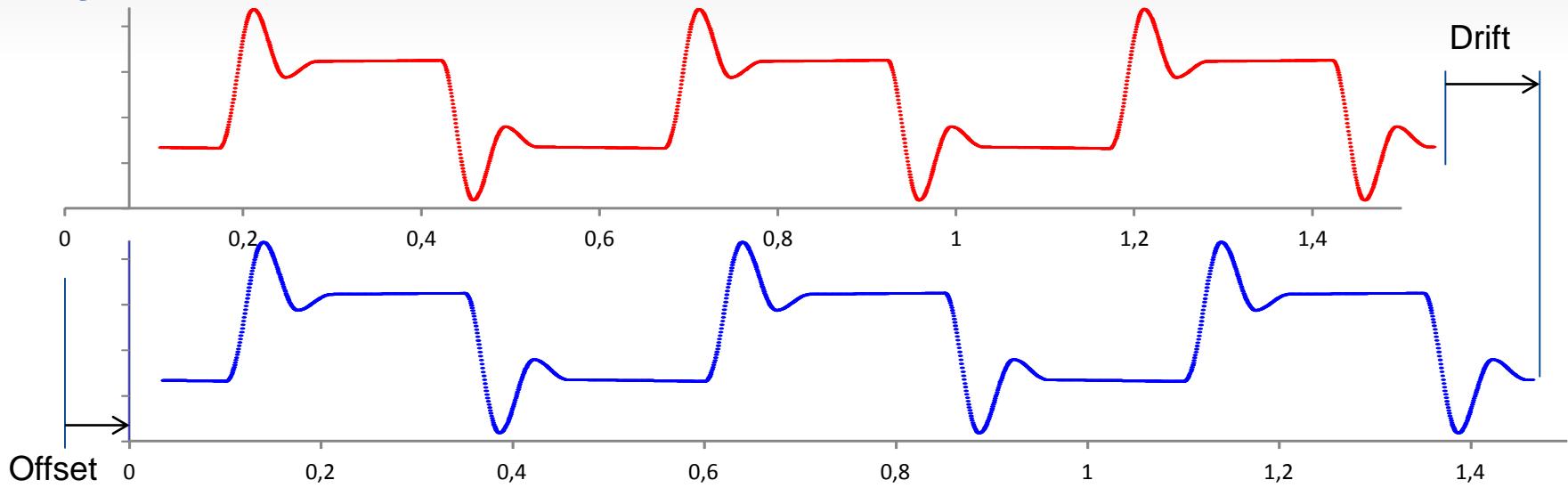
Synchronization



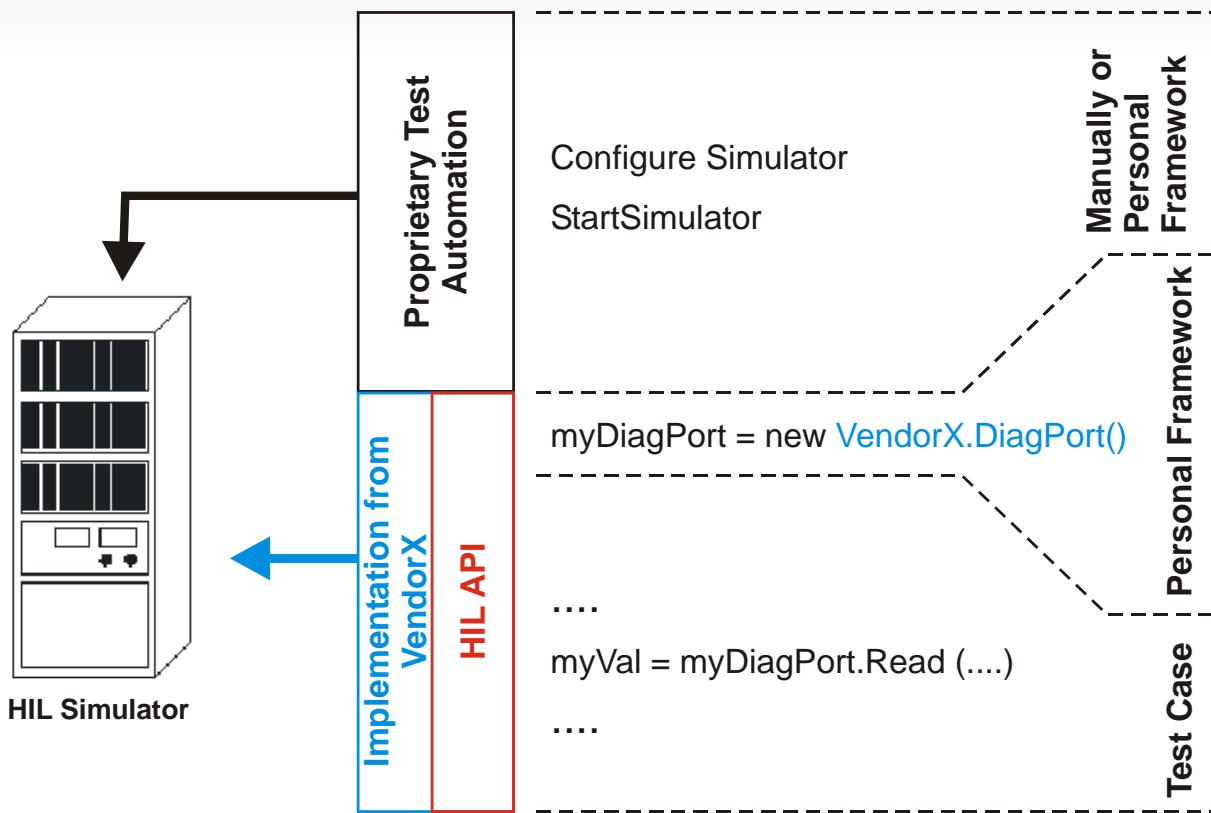
Synchronization



Synchronization

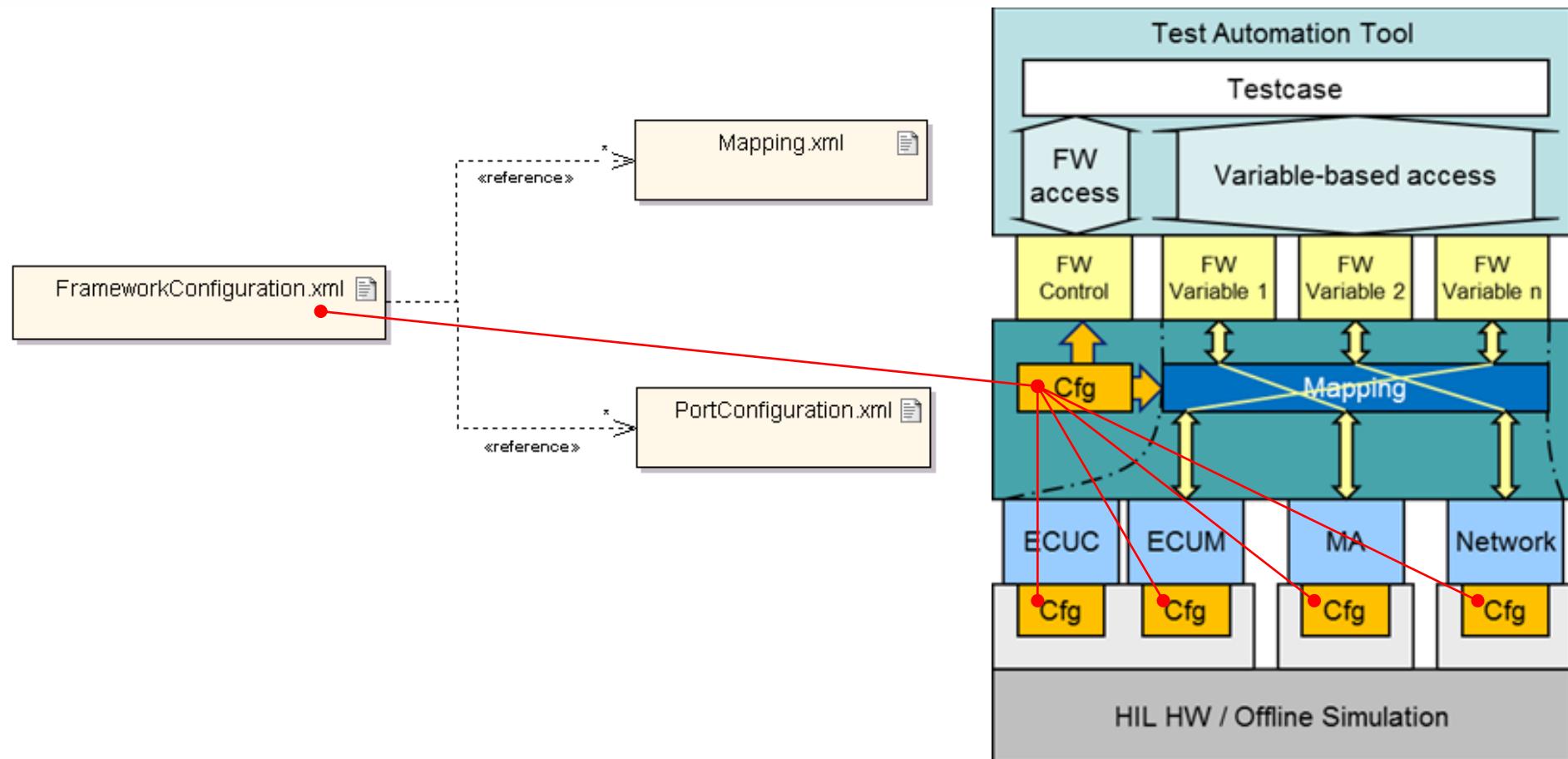


Initialization of the Simulator with HIL API 1.0.2

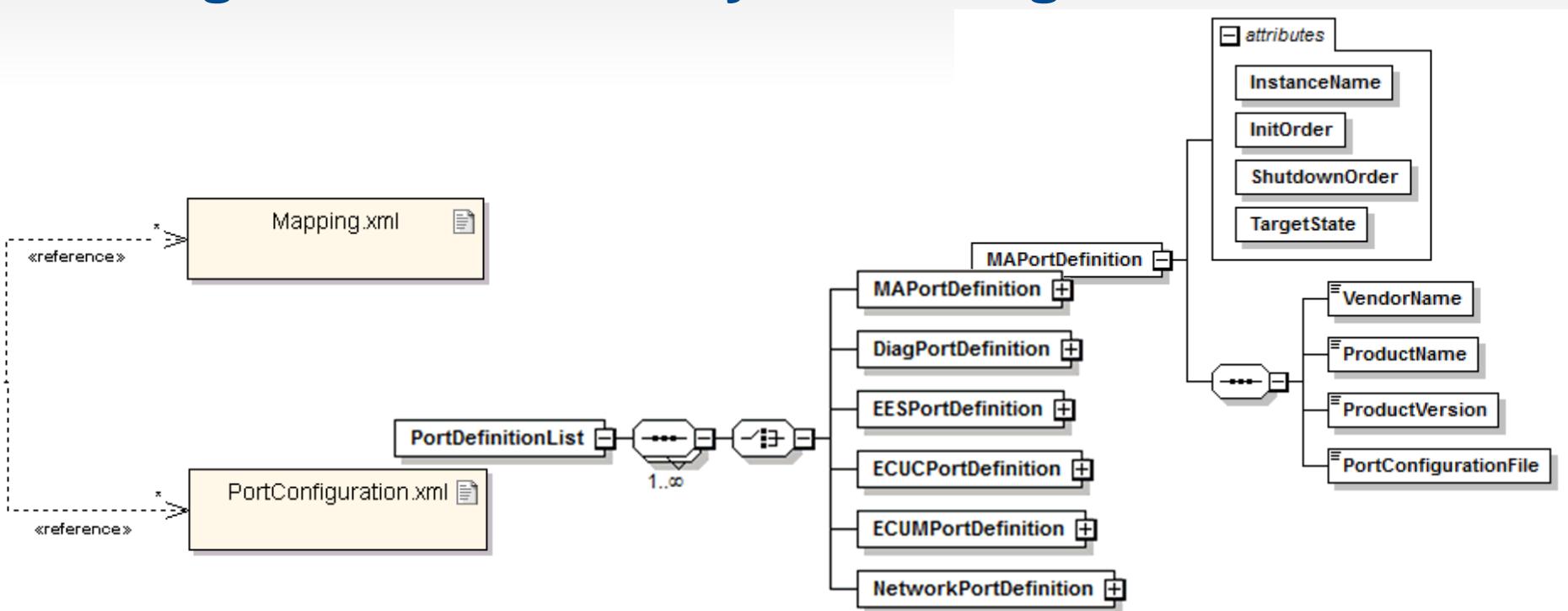


- No standardized methods available for initialization or configuration
- Thus, initialization was done manually or via personal framework

Framework Configuration with XIL API 2.0



Configuration of Life-Cycle Management



- ▶ Framework manages the port life-cycle
- ▶ Framework starts up ports in a configured order
- ▶ Framework establishes correct initial states (e. g. simulation stopped or running, online or offline, measurement stopped or running)
- ▶ Framework shuts down ports according to the shutdown order

Agenda

- | | |
|---|--------------------------------------|
| 1 | History |
| 2 | Introduction and Motivation |
| 3 | Concept of HIL API in Version 1.0 |
| 4 | Crosstests |
| 5 | Extensions of XIL API in Version 2.0 |
| 6 | Summary and Conclusion |

Summary and Conclusion

- **XIL API 2.0** comes up with broadly extended functionality:
Mapping in order to decouple test cases and test benches
Measuring of signals from different data sources with time synchronization
Managing of the test bench ports' life-cycle
- Easy test case exchange between different vendors and even between different development stages, e. g. 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.