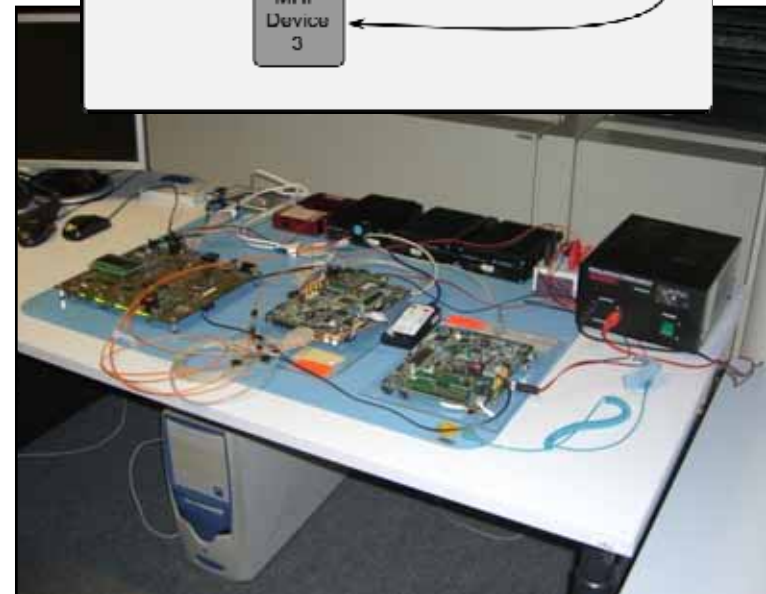
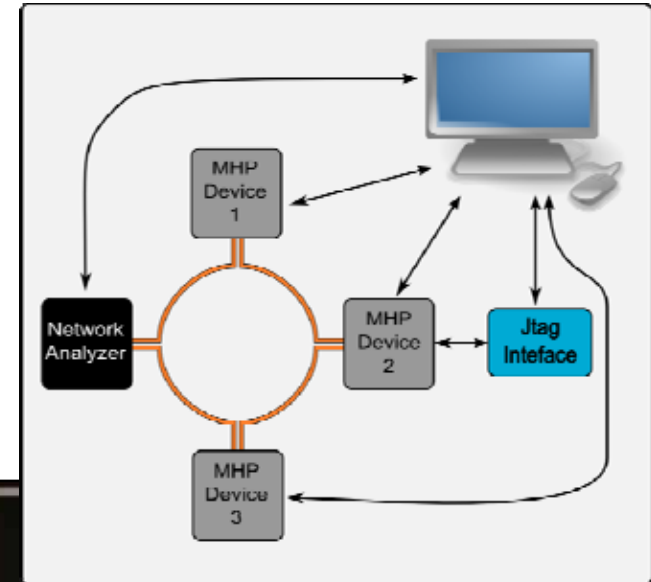


Test Description and Automation for Software Verification on Virtual Prototypes using the TTWorkbench

Andreas Braun,
Oliver Bringmann,
Wolfgang Rosenstiel

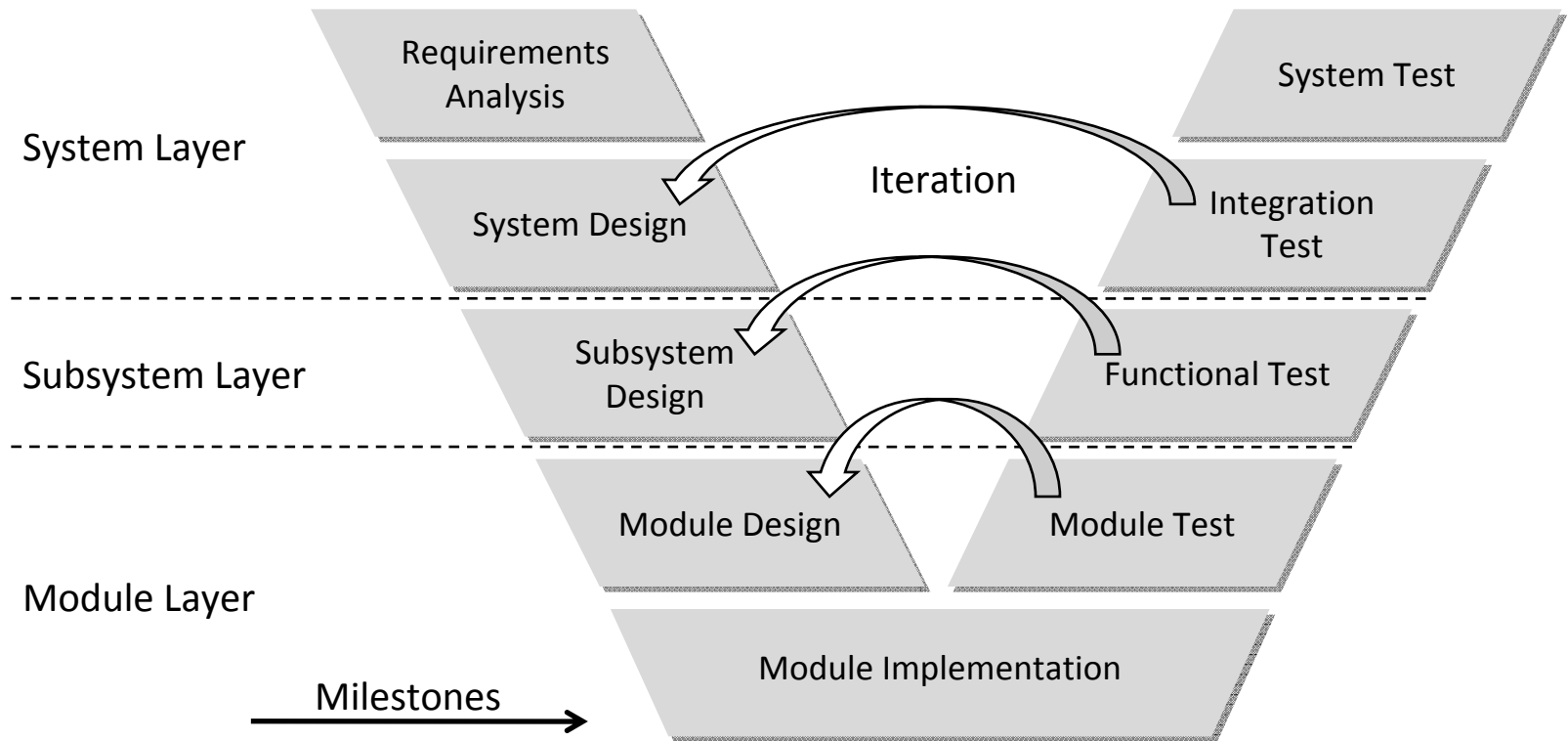
Distributed Embedded Software

- Complexity of modern distributed embedded software is rising rapidly
 - Up to 90% of innovations in cars are enabled by software
- Software verification using real prototypes is expensive
 - Need of expensive hardware
 - High effort for test configuration
 - Connecting cables, flashing software, ...
- Quality is reached by high test coverage
 - Future software verification using real hardware requires additional hardware and engineers



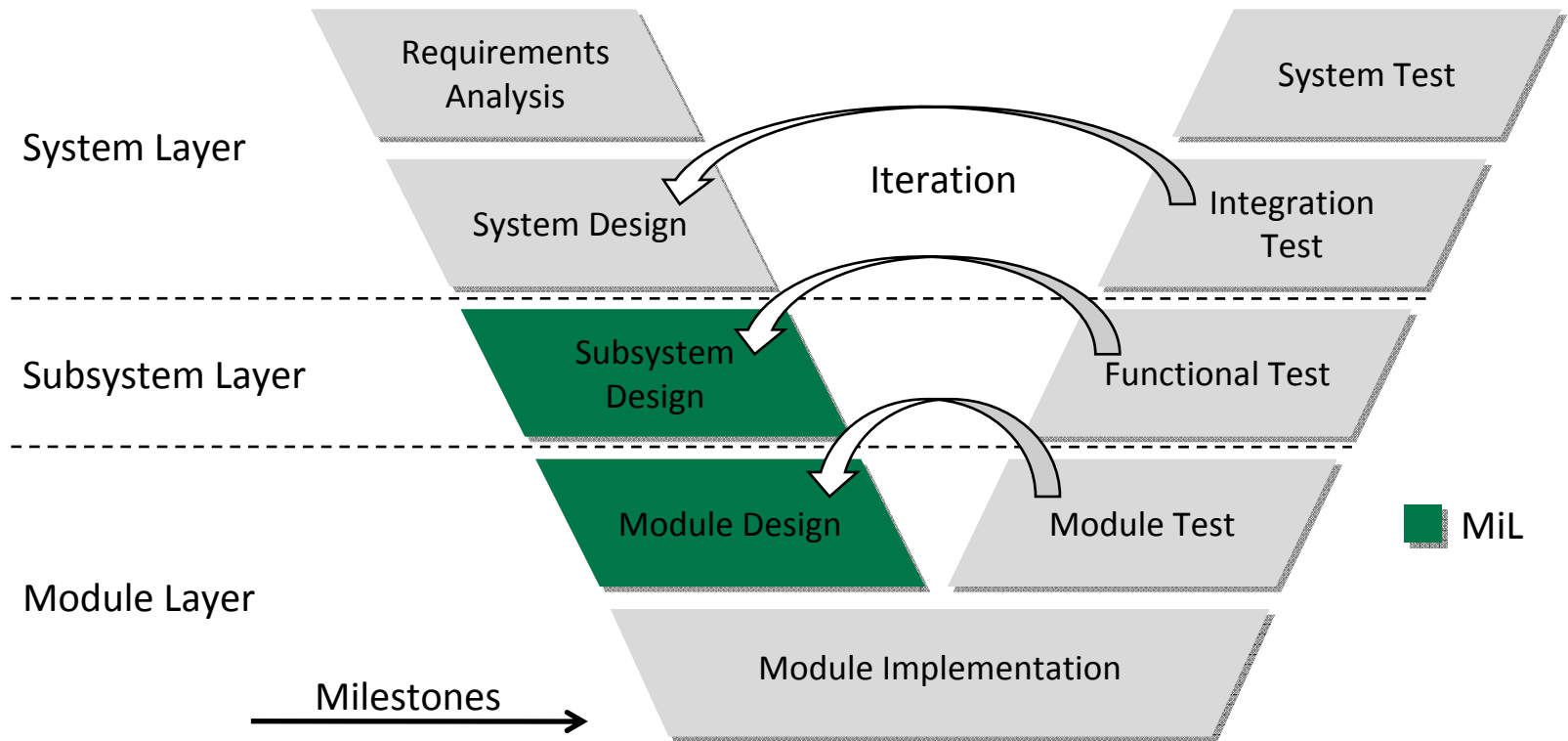
Source: SMSC Europe GmbH

Software Development using the V-Model



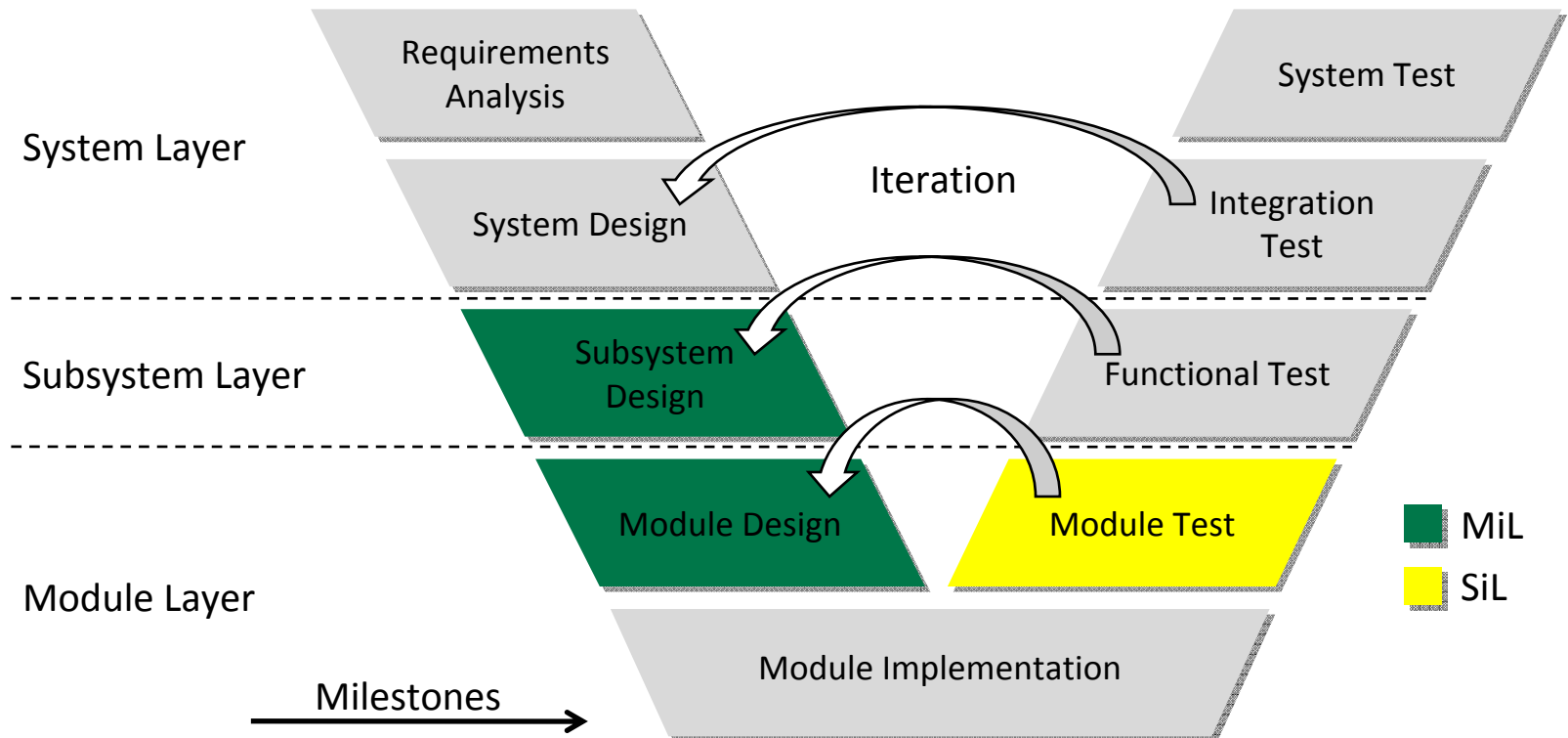
- Design iterations across different phases are expensive
 - Early verification avoids iterations across different phases

Software Verification using the V-Model



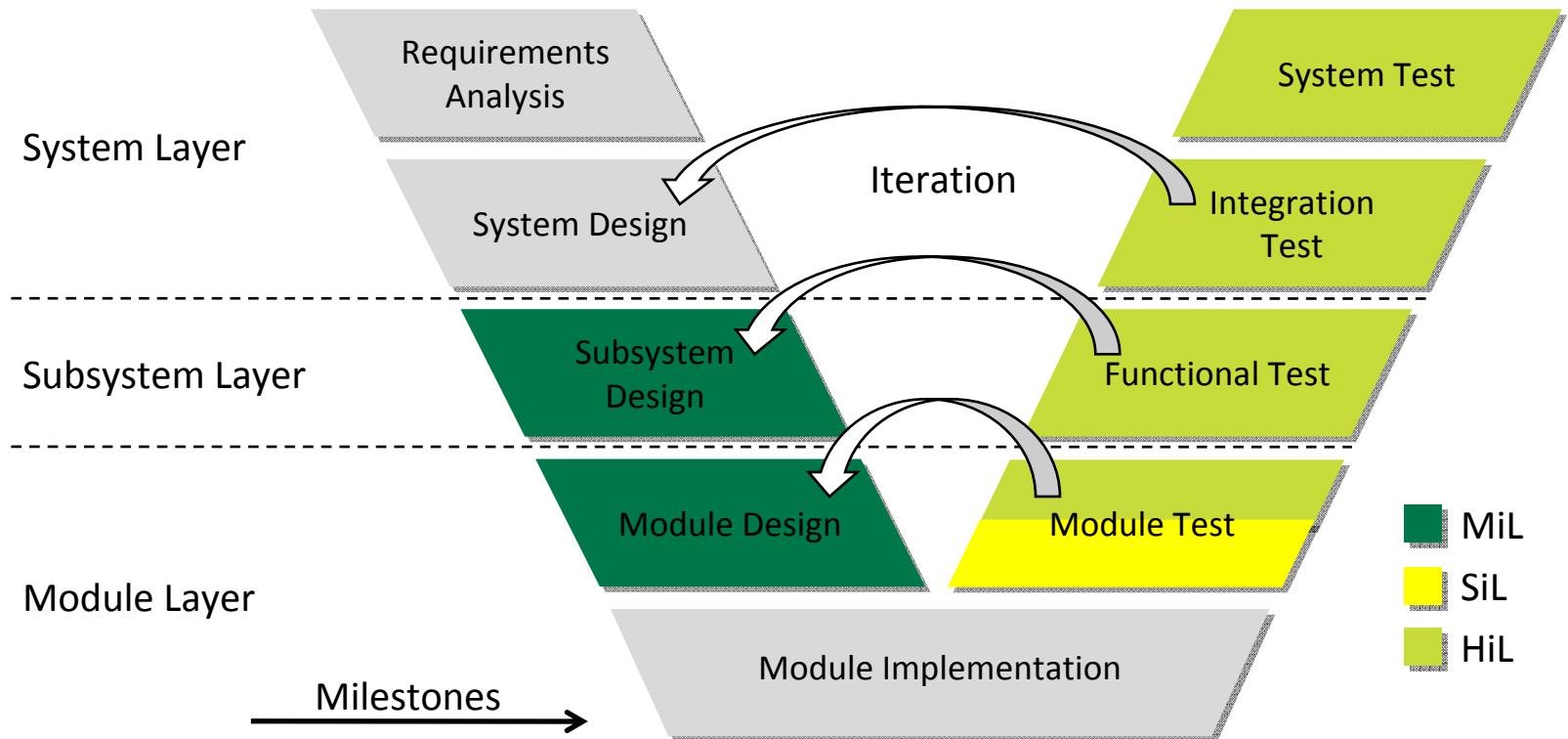
- Model in the Loop (MiL)
 - Verification during design phase
 - Applicable to subsystem and module layer

Software Verification using the V-Model



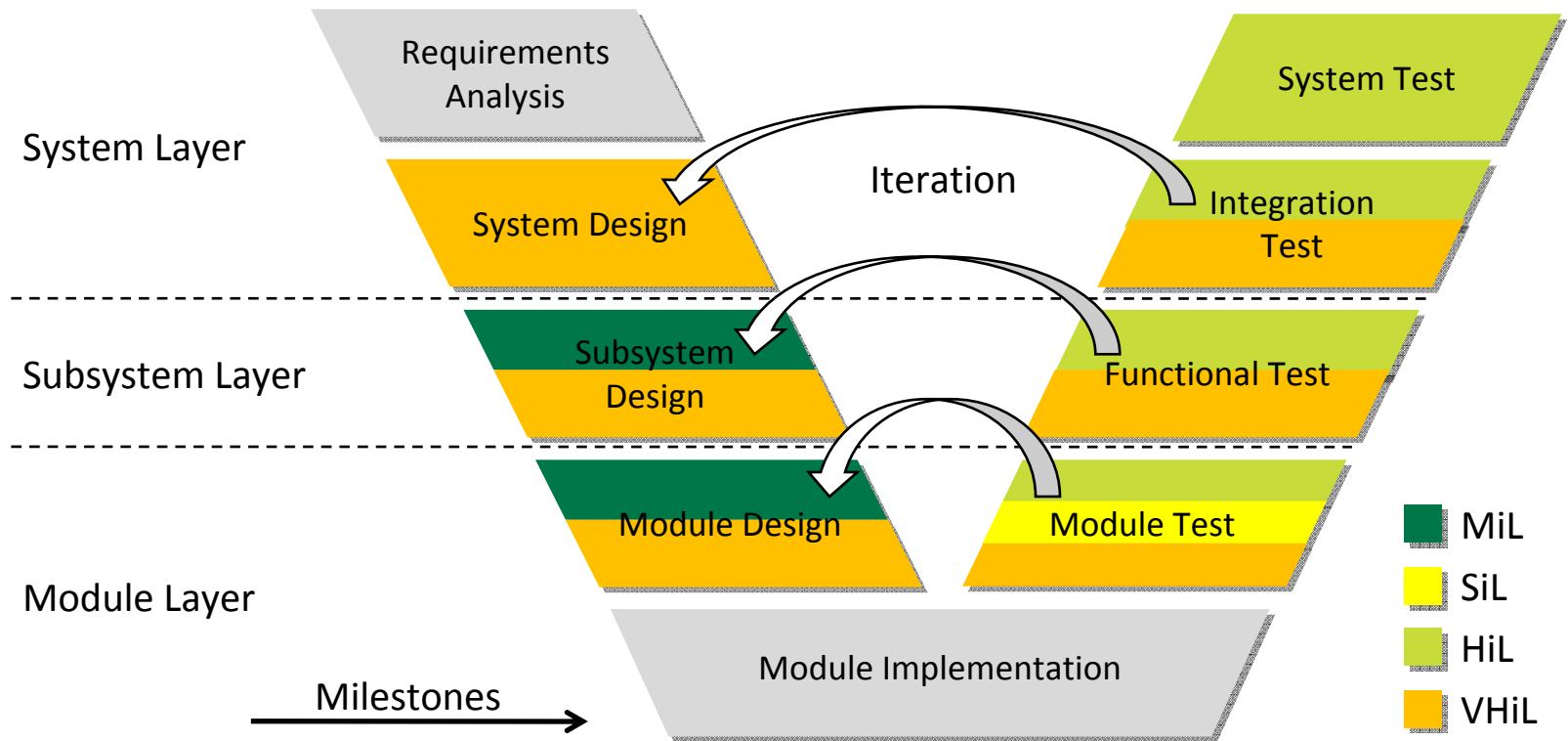
- Software in the Loop (SiL)
 - Software verification without hardware consideration
 - Applicable to module layer

Software Verification using the V-Model



- Hardware in the Loop (HiL)
 - Software verification using real prototypes
 - High grade of test automation
 - Applicable to module, subsystem and system layer

Software Verification using the V-Model



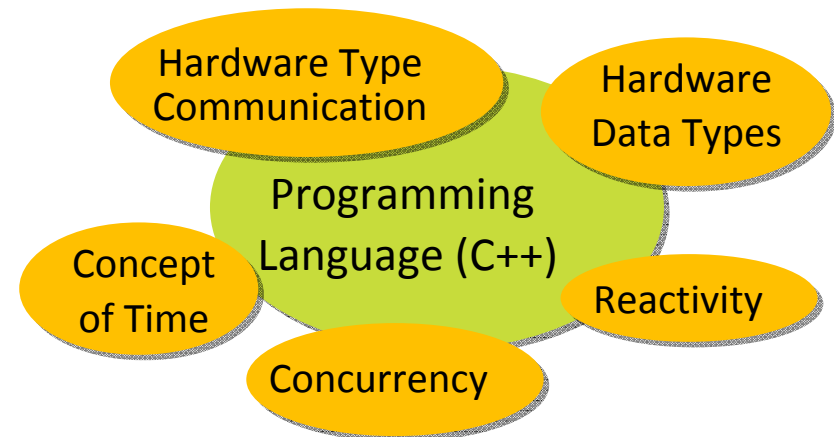
- Virtual Hardware in the Loop (VHiL)
 - Software verification using virtual prototypes
 - Applicable to module, subsystem and system layer
 - Tests in early phases can avoid expensive iterations
 - Tests in later phases can increase test coverage

- Virtual Prototype for MHP Simulation
 - Connection of the TTWorkbench
 - Conclusion

System Modeling and Verification with SystemC



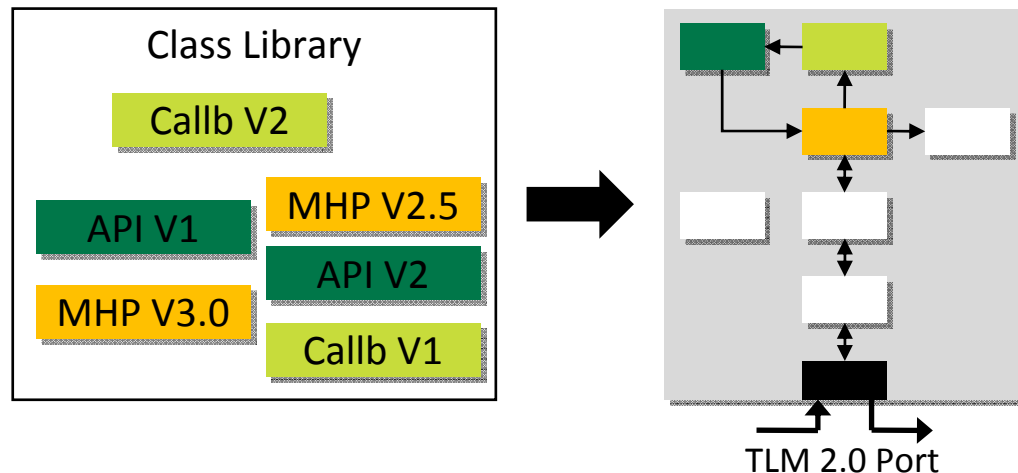
- SystemC is a C++ library which provides a set of routines and macros for system modeling
 - Allows simulation of accurate timing and functional behavior at system level
 - Allows modeling of concurrency for different modules
 - Modeling of different abstraction levels
- SystemC Verification Library (SCV)
 - Provides routines and macros for verification
 - Data introspection
 - Pseudo random generation



Solution: Class Library

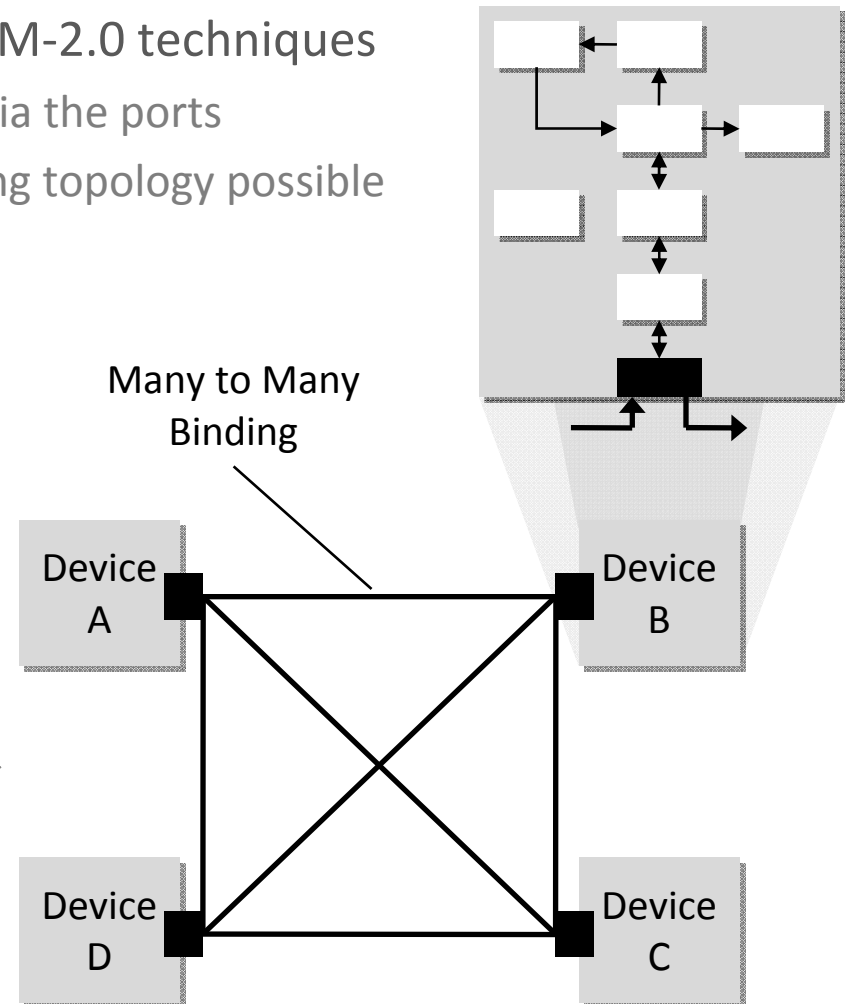
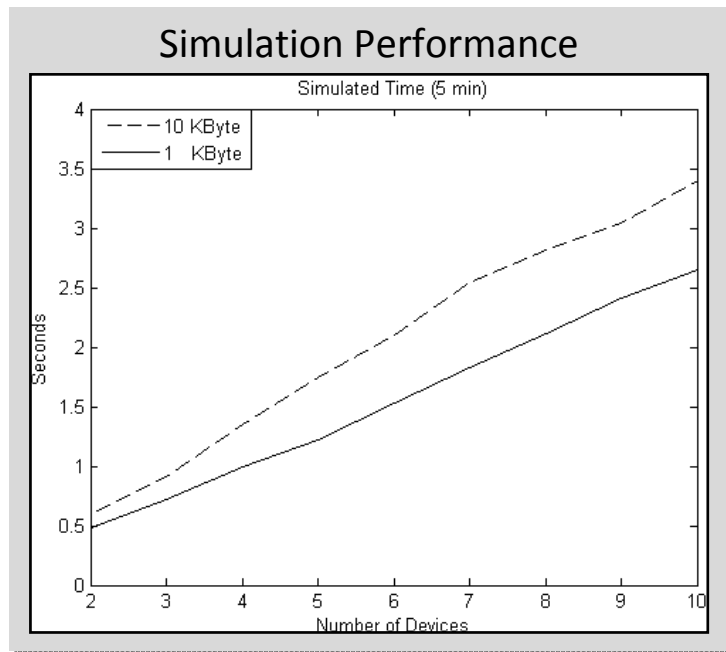
MHP Virtual Device Modeling

- MOST High Protocol (MHP) protocol stack is written in C
 - Encapsulation in C++ class allows multiple instantiation of the protocol stack
- Devices can be built from different elements with different functionality
 - Modules are instantiated from an integrated library
 - Library can be easily extended
 - Platform configuration is done by parsing an XML file

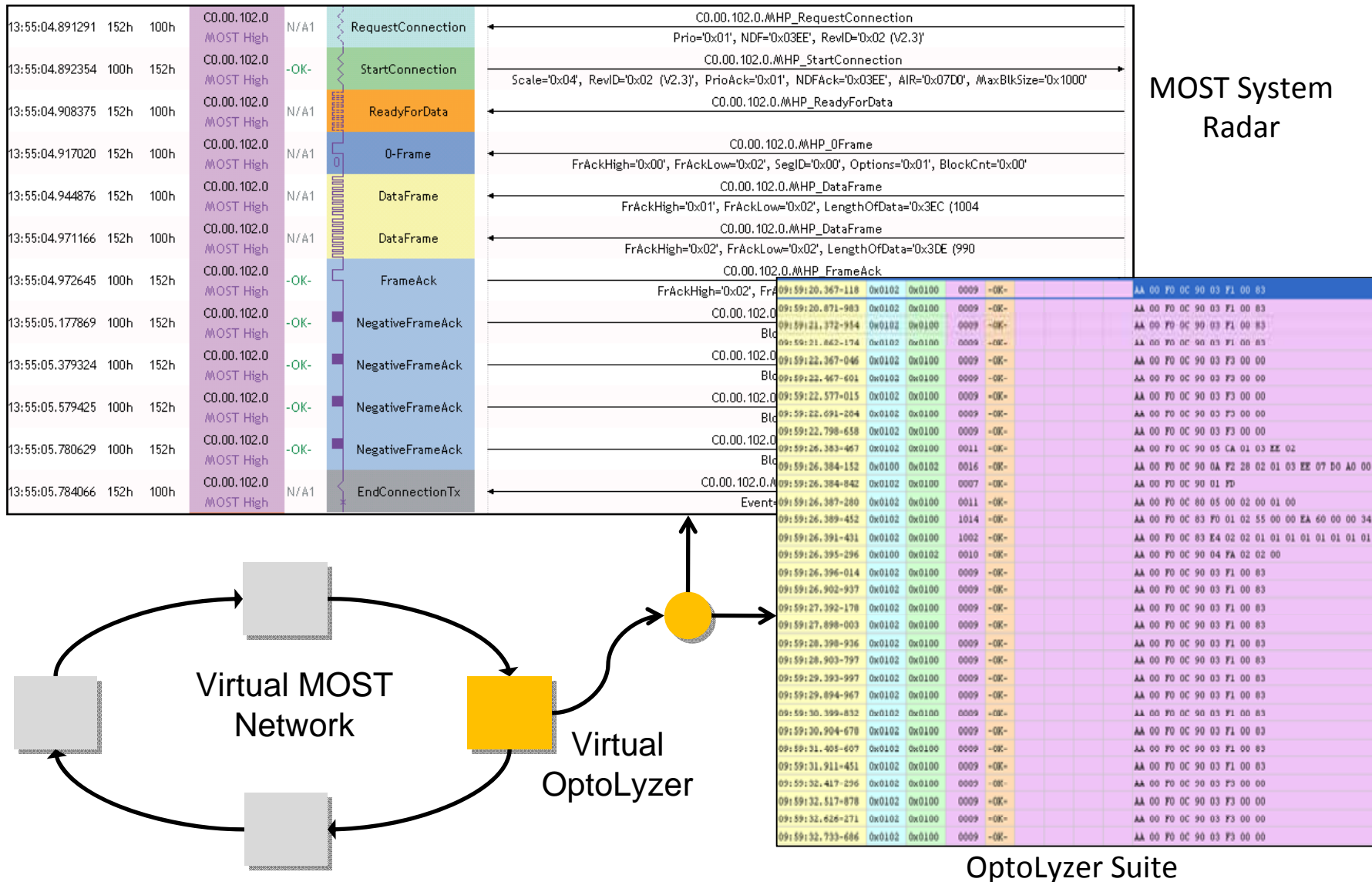


MHP Virtual Network Simulation

- Devices are interconnected using TLM-2.0 techniques
 - MHP data structure is transmitted via the ports
 - Many to many binding instead of ring topology possible
 - Simulation acceleration



Test Visualization with OptoLyzer Suite and MOST System Radar

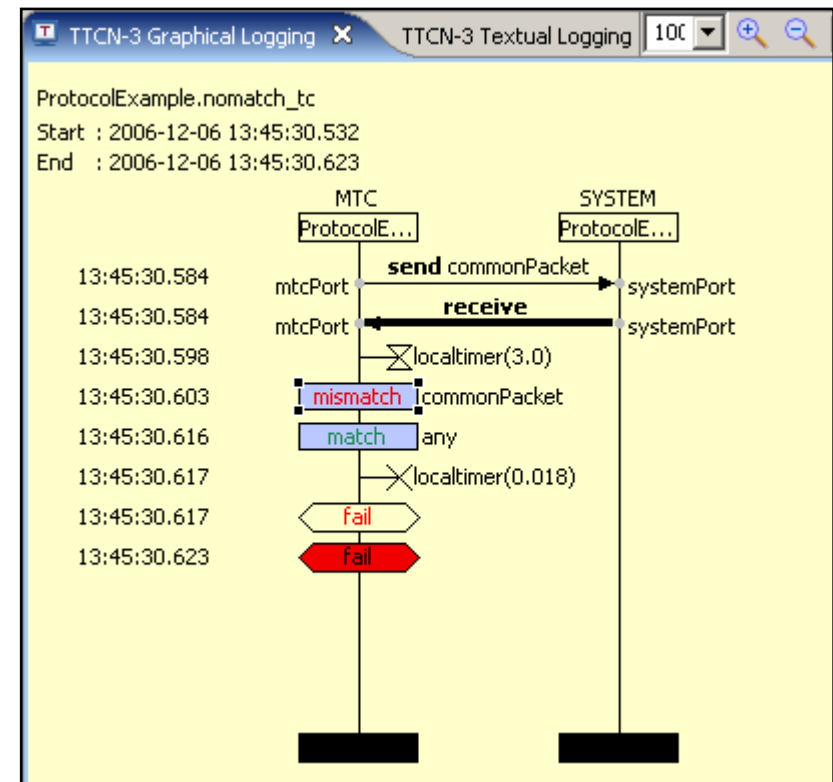


MOST System Radar

- Virtual Prototype for MHP Simulation
- Connection of the TTWorkbench
- Conclusion

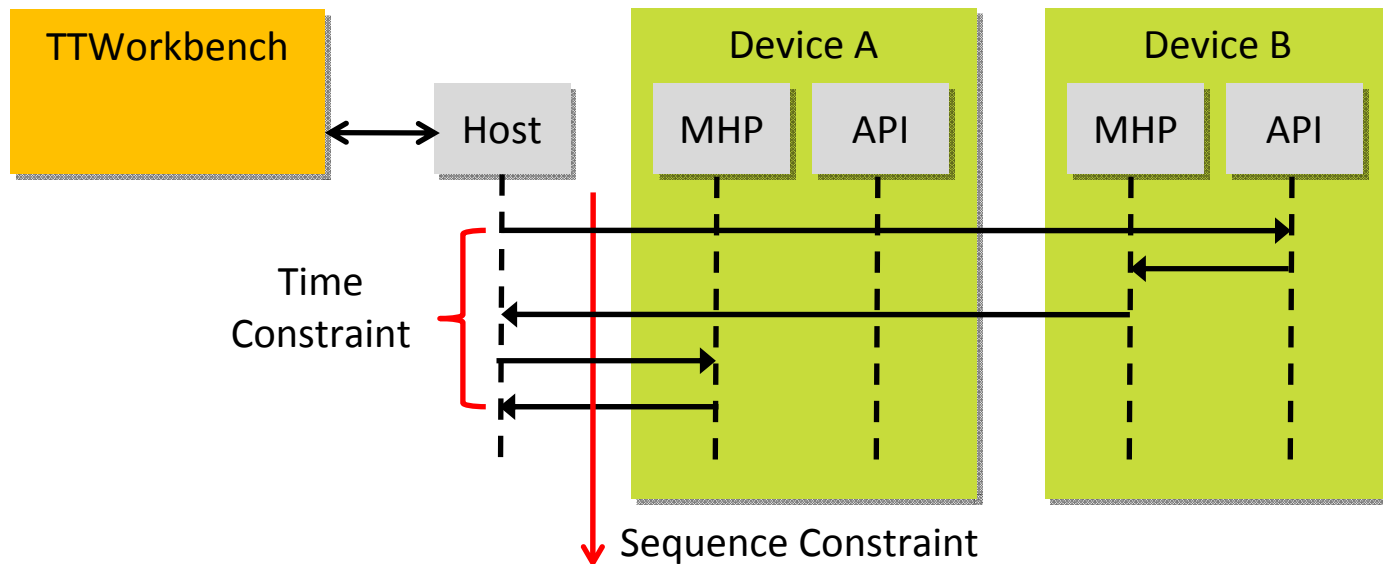
- TTWorkbench (Testing Technologies)
 - Allows transaction based testing (function oriented)
 - Known tool for protocol testing using real hardware (MOST MHP)
 - Textual description using Testing and Test Control Notation (TTCN-3)
 - Visualization of test results using sequence diagrams

- No consideration of platform configuration
 - Loading of XML platform config.
 - Useful for automatic HiL config.



Host Based Verification

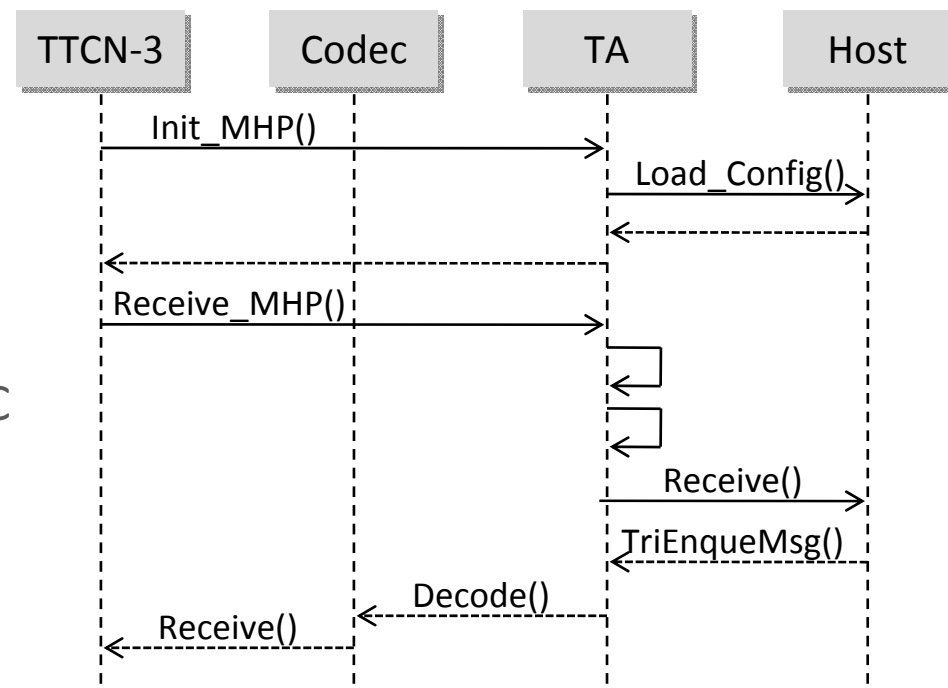
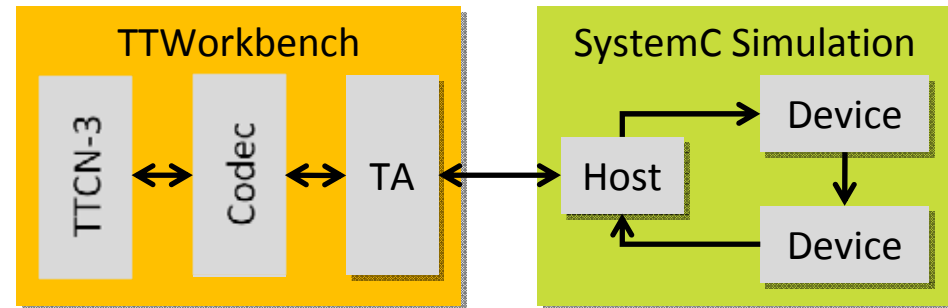
- For connection to the TTWorkbench a host module was developed
 - Activation and forwarding of message transmissions
 - Allows module, integration and regression testing
- TTWorkbench controls the virtual host and controls the test
 - Checking of time and sequence constraints



Connection of TTWorkbench and SystemC



- TTWorkbench controls SystemC simulation
 - Init_MHP()
 - Loading of platform configuration
 - Send_MHP()
 - Writing message in queue
 - Receive_MHP()
 - Waiting until message receives
 - Triggering of simulation model
 - Wait(), Stop_Simulation(), ...
- Time synchronisation with SystemC
 - Synchronization with the TTWorkbench via the TA
 - SystemC acts as timing master



- Virtual Prototype for MHP Simulation
- Connection of the TTWorkbench
- Conclusion

Usage of the MHP Simulation



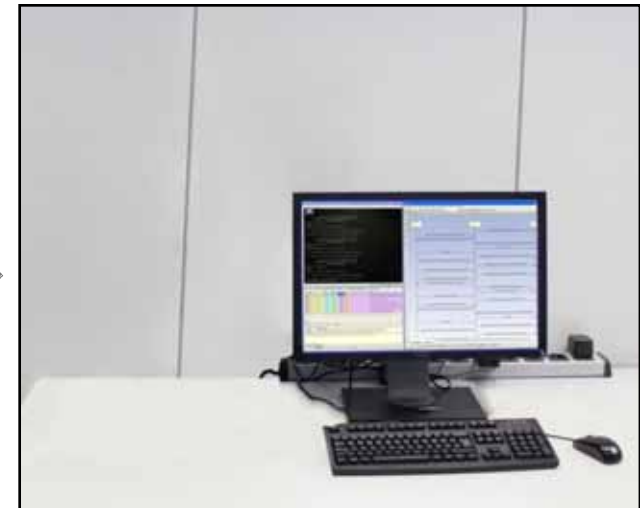
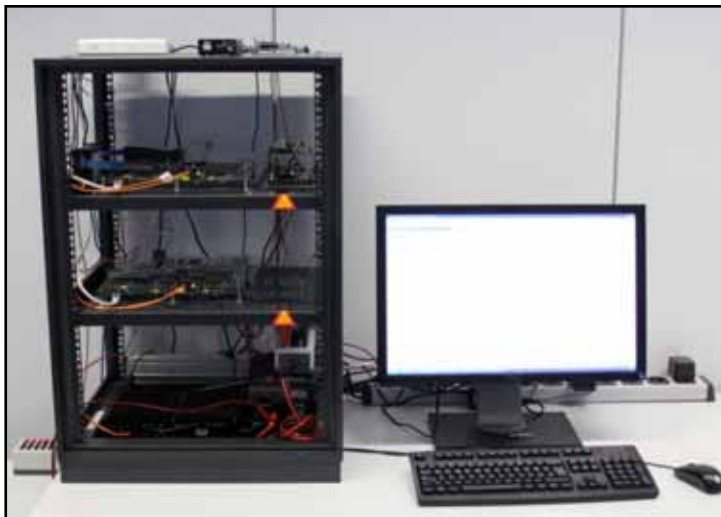
- Application development and testing (FZI)
 - Automatic application testing using multiple configurations
- Performance evaluation (FZI)
- Platform evaluation (FZI/ SMSC)
 - Design space exploration by testing multiple configurations
- Testing of error situations (SMSC)
 - Loss of messages, transmission of wrong parameters, ...
- Regression testing (SMSC)
 - Early and fast verification using predefined test cases
- Support case evaluation (SMSC)
 - Customer problems can be reproduced by the model
 - No real hardware connection, high traceability, ...



SMSC[®]
SUCCESS BY DESIGN

Combined Verification on Real and Virtual Prototypes

- Verification on real and virtual hardware complement each other
 - Test cases are reusable on virtual and real hardware
 - High test coverage reachable using virtual and real prototypes
 - Automatic platform configuration, high simulation speed, ...
 - Testing in early and in later phases possible
- Additional effort for virtual prototype development



Source: SMSC Europe GmbH

Thank you very much for your attention

abraun@fzi.de

bringmann@fzi.de