Joint Research Activity 2: Co-Simulation Based Assessment Methods

1st April 2020

Peter Palensky, TU Delft

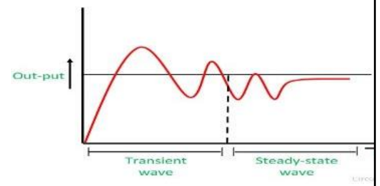
ERIGrid Final General Assembly

Vienna, Austria

Contents

- Motivation
- Vision & Challenges
- JRA 2 Approach
- Co-simulation assessment for continuous-time RMS studies (TC 1)
- Combined Hardware and Software Simulation (TC 2)
- Signal-based Synchronization between Simulators (TC 3)
- Major Achievements
- Deliverables & Dissemination

Motivation



Physics

continuous process

energy generation, transport, distribution, consumption, etc.

Roles

behavioral process

agents, game theory, market players, etc.

cyber-physical energy system

+ Speed!
+ Size!
+ ...

Information Technology

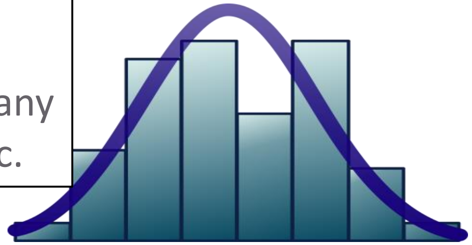
discrete process

controllers, communication infrastructure, software, etc.

Stochastics

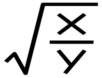
statistical process

weather, aggregate of many individual elements, etc.

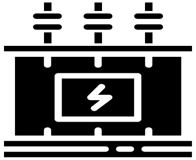


Chapter 3.1 of European guide to power system testing

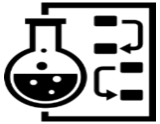
Analysis options for CPES



Analytical → too complex



Real component testing → live patient



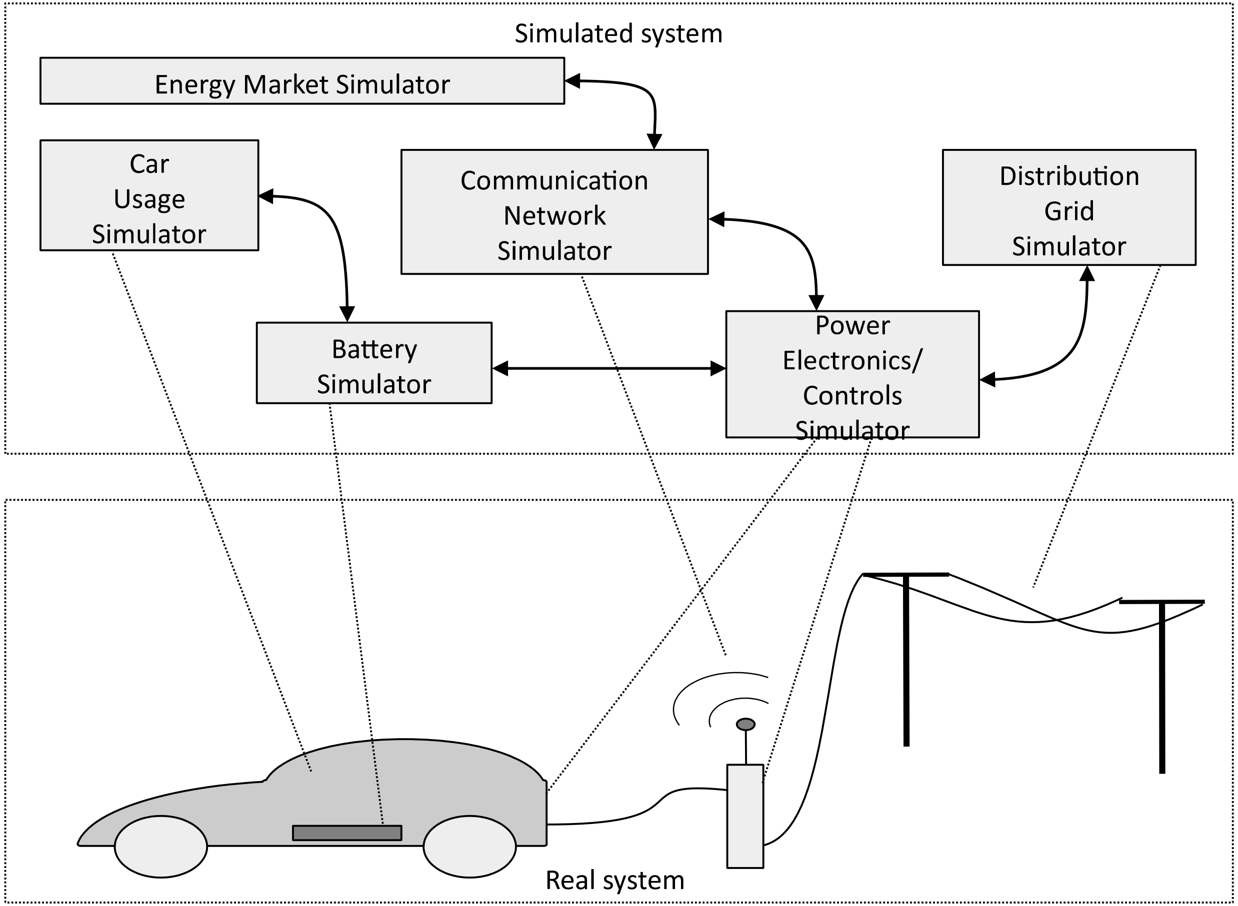
Laboratory testing → System under Test limited size

(Real-Time-)Simulation, maybe with Hardware-in-the-loop → which software?

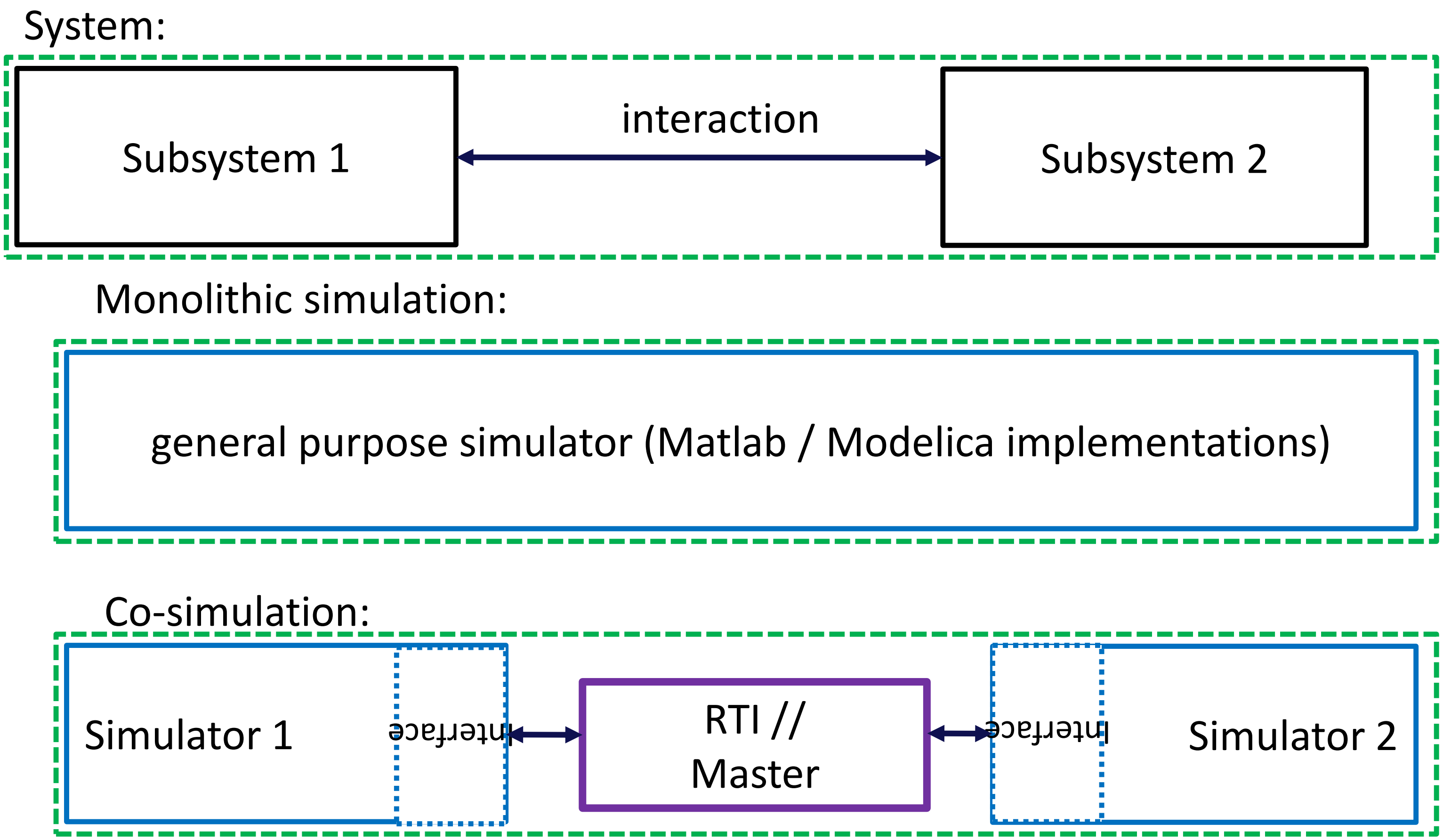
EU H2020 Programme GA No. 654113

Multi-disciplinary simulation

Coupling Simulators for a connected world



“The" solution: co-simulations



Use specialised tools
Standardised interfaces
Good accuracy
Good performance

Implementation
Scalability
aaS

RTI: runtime environment

JRA2 Approach

Development of 3 test cases to assess the different smart grid behavioral phenomena by means of **co-simulation**:

- Test Case 1 (TC1): to investigate the *cyclic dependencies* between continuous simulators
- Test Case 2 (TC 2): to investigate *combined hardware and software simulation*
- Test Case 3 (TC 3): to investigate *signal-based synchronisation* between simulators

To supplement the co-simulation framework, further research was conducted for:

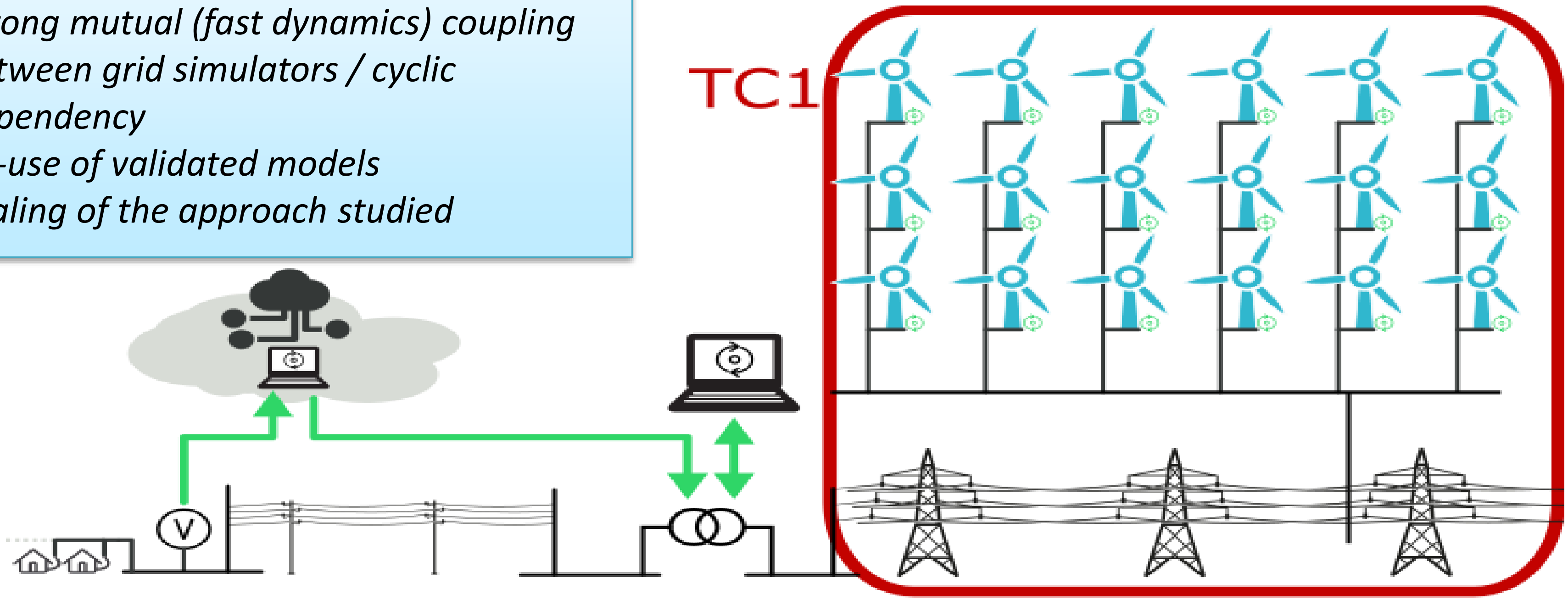
- Development of relevant coupling tools and interfaces for different simulation platforms
- Development of model libraries to cover the state-of-the-art of smart grid technology

Co-simulation assessment for continuous-time power system studies (TC1)

Chapter 3.5 of European guide to power system testing

Grid+wind park & FRT

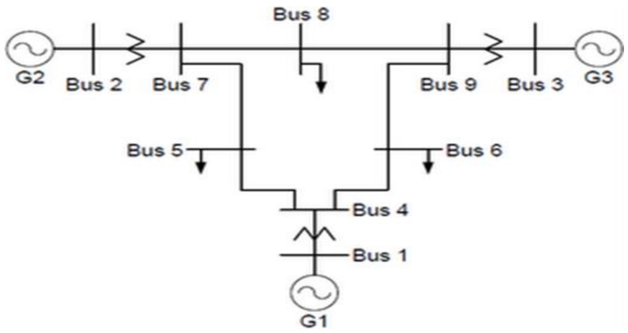
- *Strong mutual (fast dynamics) coupling between grid simulators / cyclic dependency*
- *Re-use of validated models*
- *Scaling of the approach studied*

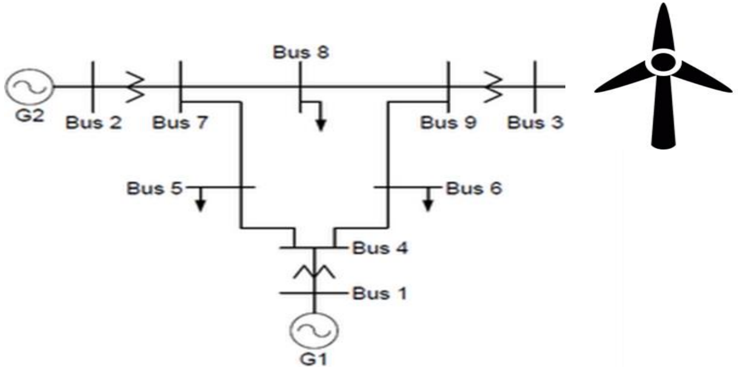


FRT: Fault-Ride-Through

Models and test system rationale

grid model that exhibits the tight coupling between its subsystems and allows rapid prototyping

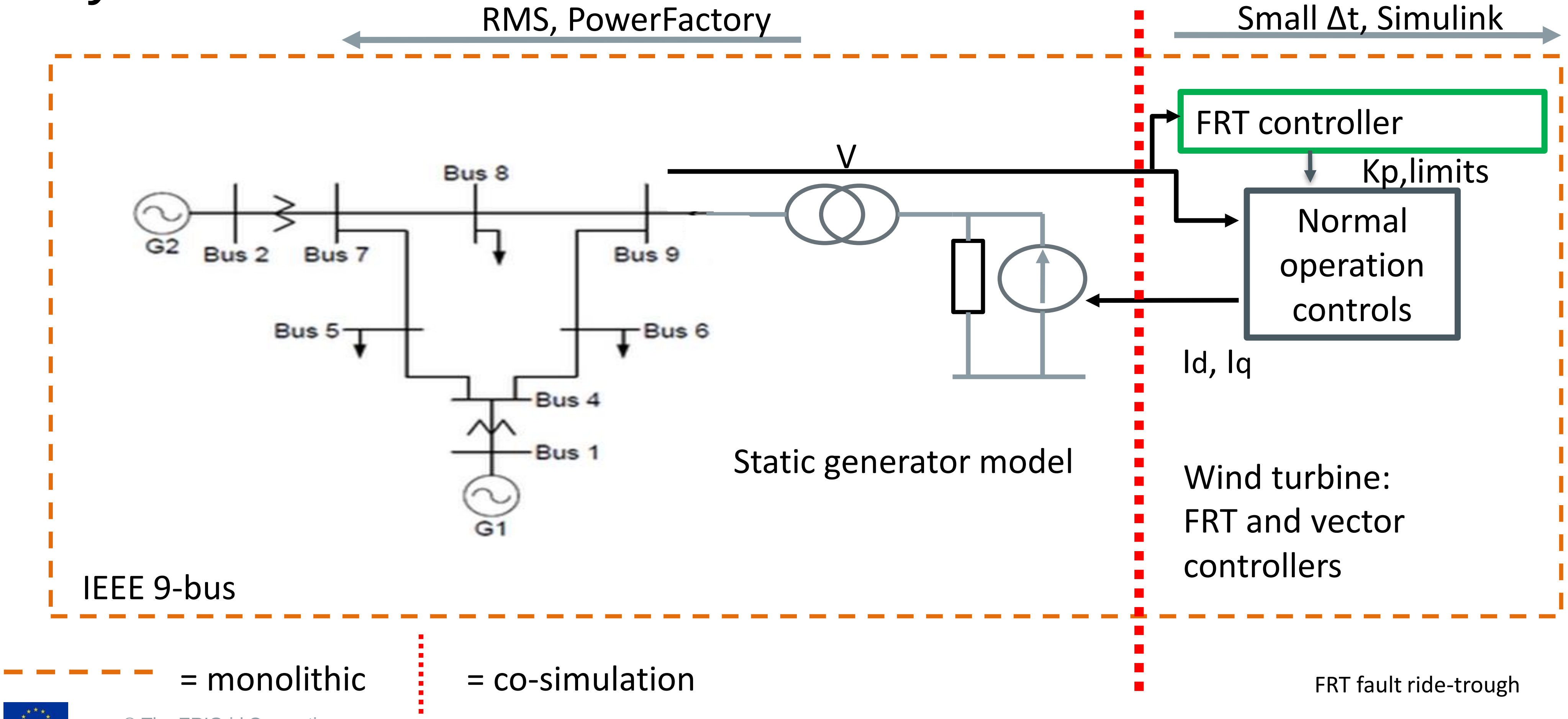




- IEEE 9-bus system
- Commonly used for systematic analysis of transient stability concepts
- G3 replaced by a type IV wind turbine

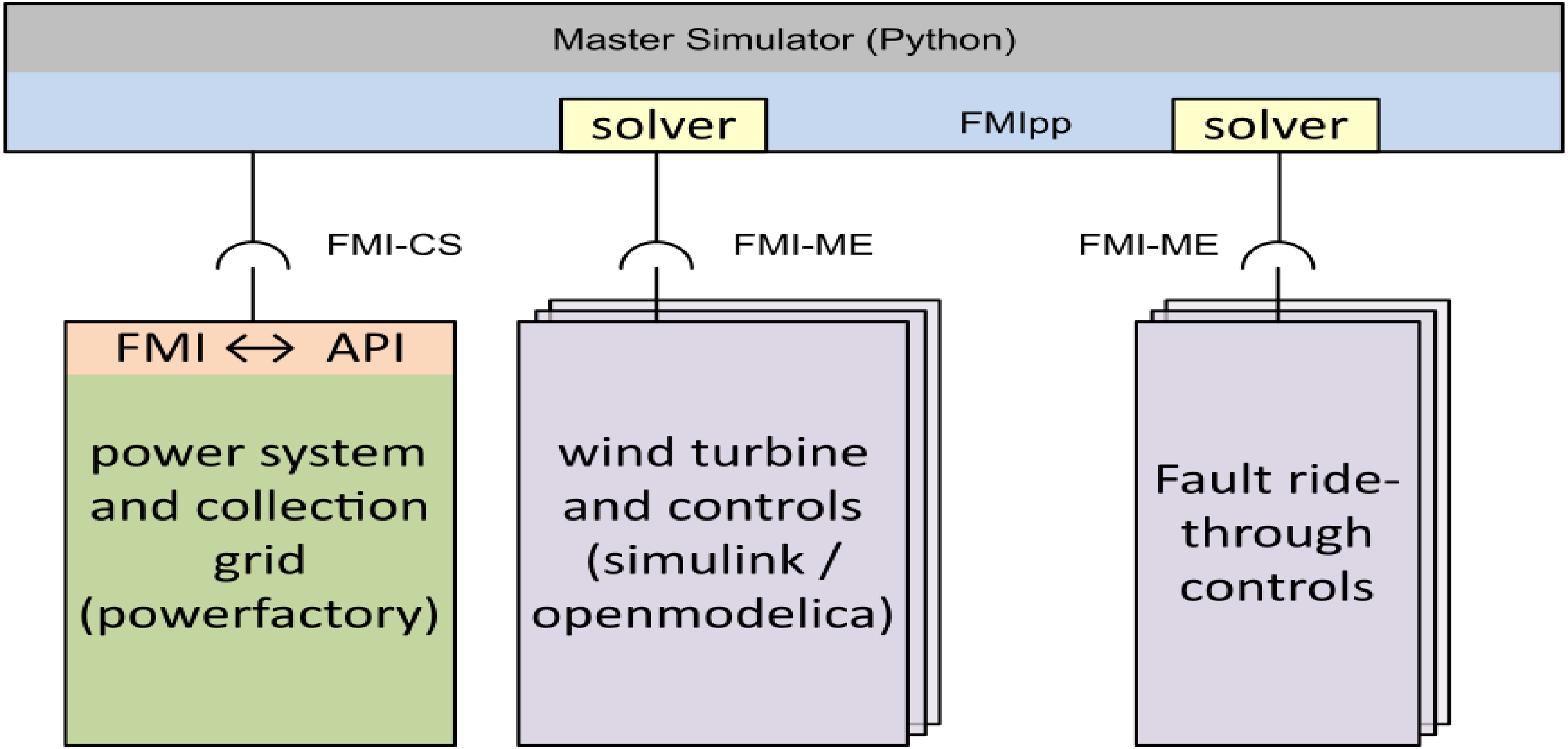
- Wind turbine model according to IEC 61400-27-1
  - Type 4 wind turbines: grid interaction dictated by control scheme
  - dc-side, aerodynamic part, mechanical part abstracted away
  - Simplified representation for stability studies
  - Grid interface: Norton current injection

System under Test



FRT fault ride-trough

Co-simulation experiment setup



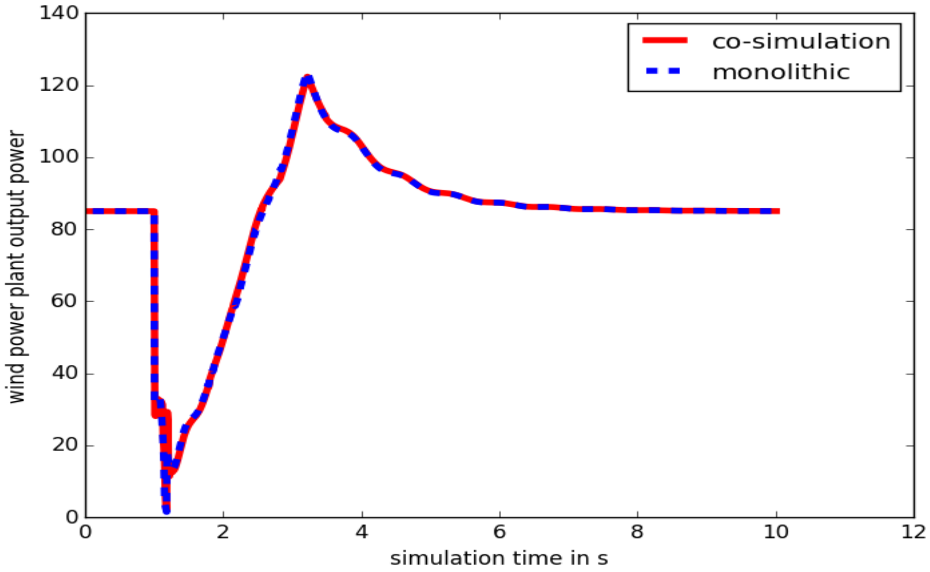
Co-simulation Testing

Test Name	Platform	Purpose	Modifications
Monolithic	PowerFactory	Reference simulation	Gen. G3 in IEEE 9 Bus replaced by WPP
Small Scale Co-Simulation	PF & Matlab & FMI++	Simple co-sim for assessment	No model modifications
Large Scale Co-Simulation	PF & Matlab & FMI++	Co-sim performance check for complex situations and numerically bigger systems	WPP divided in 32 smaller WTGs to have realistic representation. Similarly 32 added converter and FRT controllers.

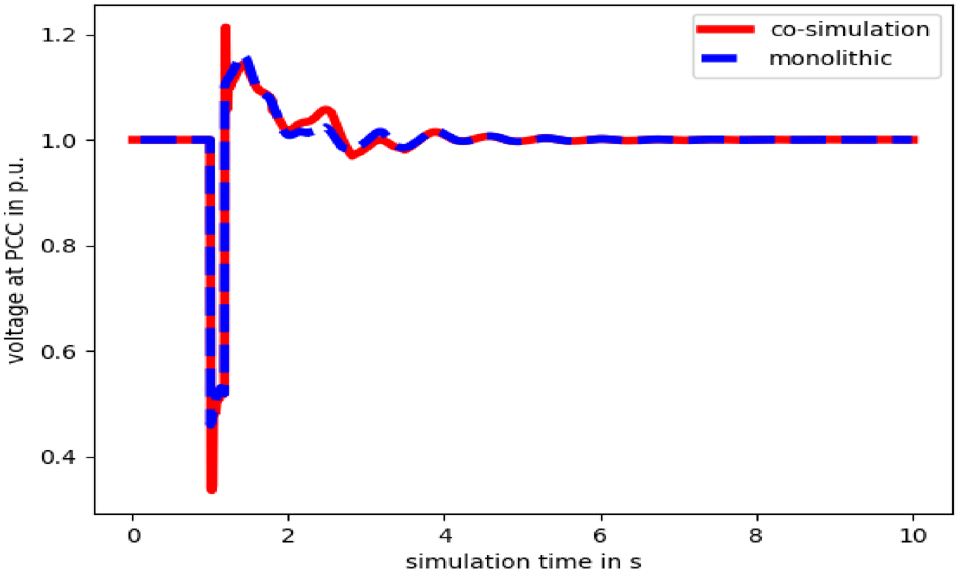
PF: powerfactory
WPP: wind power plant
WTG: wind turbine generator

Results co-simulation vs monolithic

Output Power Comparison at PCC*



Voltage Comparison at PCC*



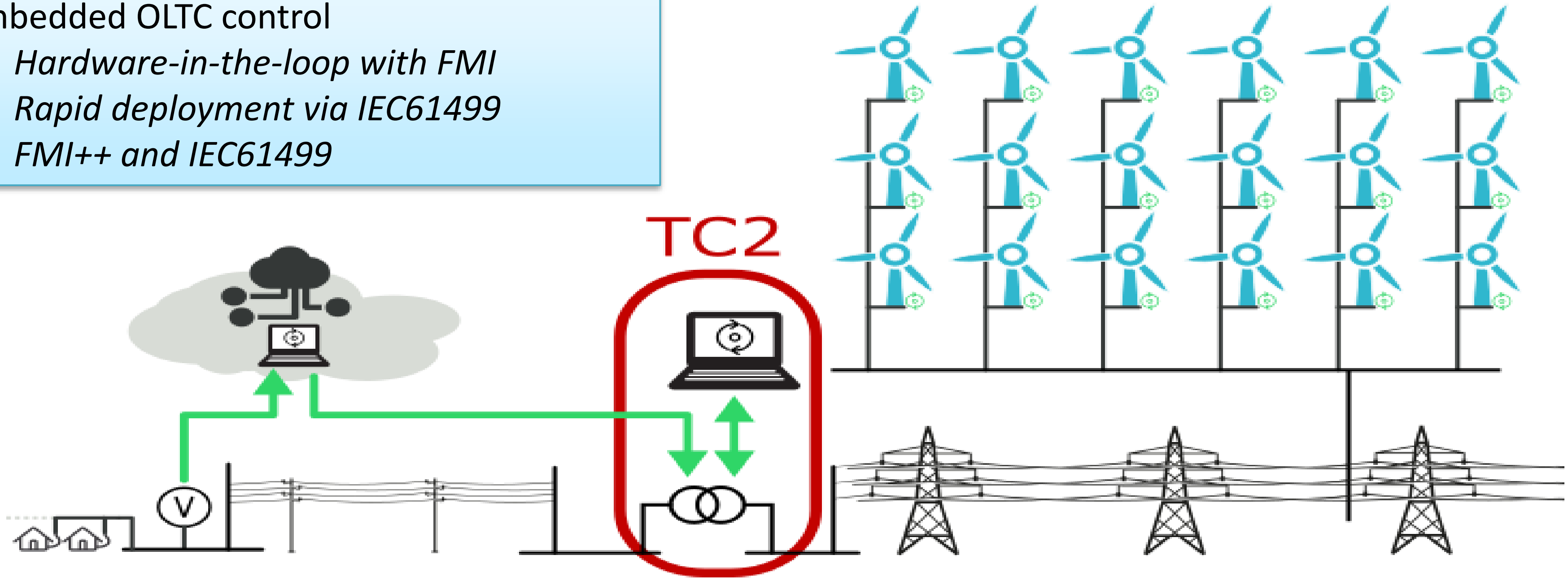
*PCC: Point of Common Coupling

Sync step size 10ms

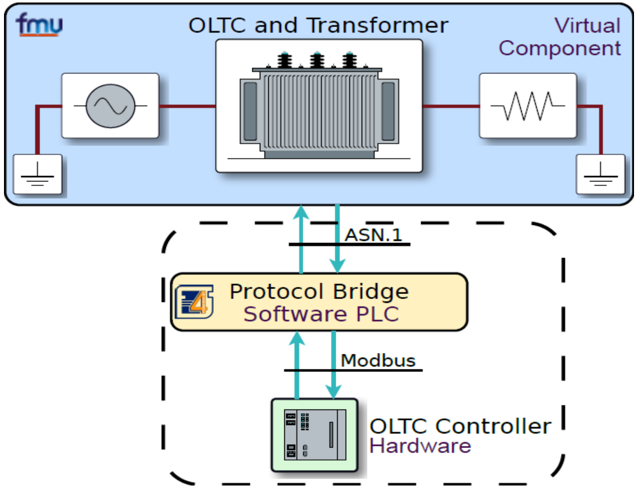
Combined Hardware and Software Simulation (TC-2)

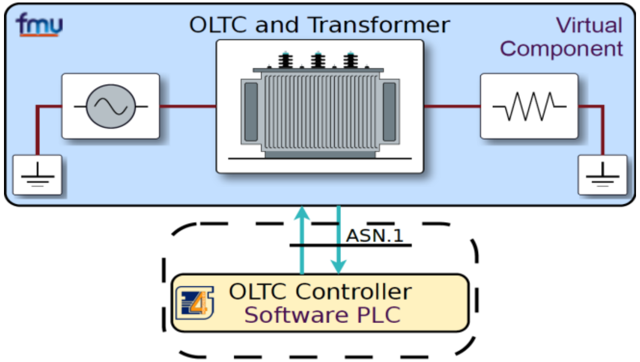
Embedded OLTC control

- *Hardware-in-the-loop with FMI*
- *Rapid deployment via IEC61499*
- *FMI++ and IEC61499*



Cases addressed





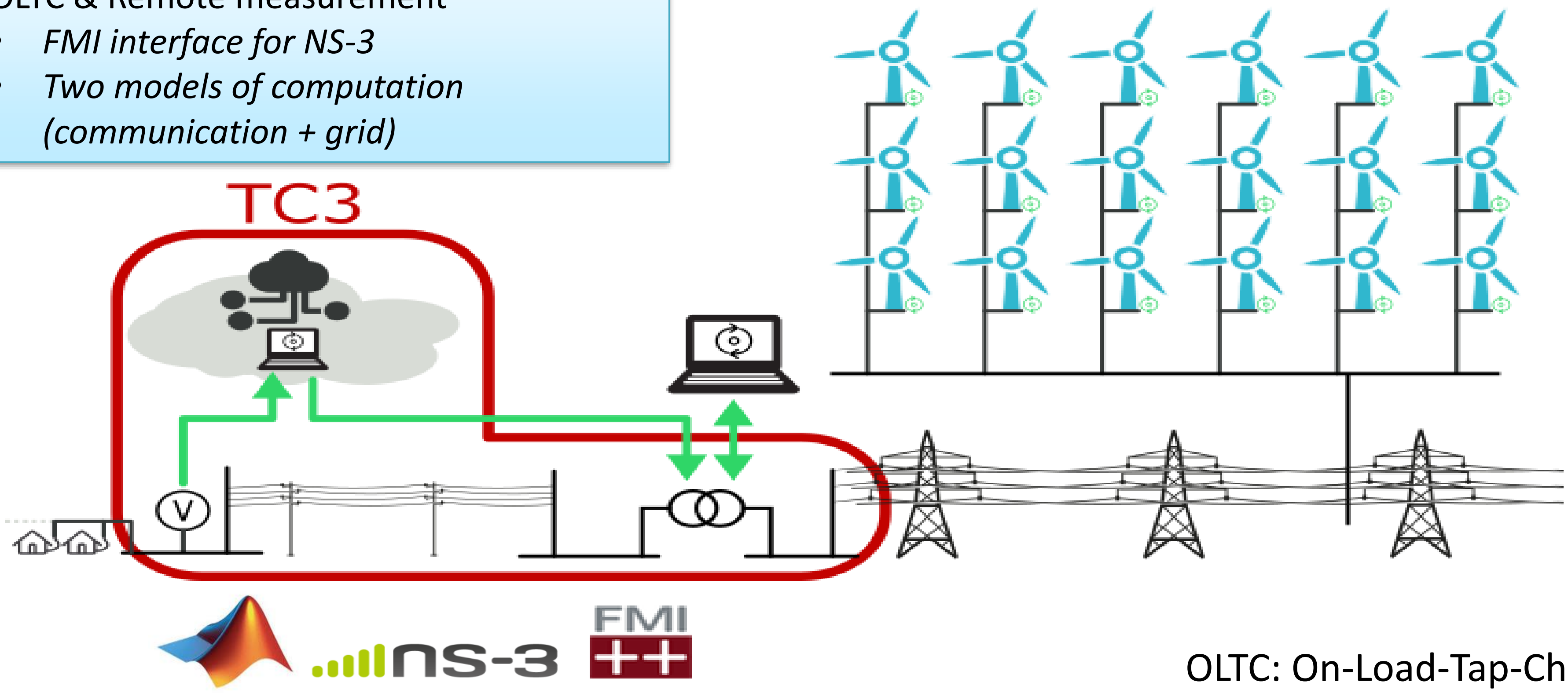
Controller emulated on an Arduino, Communication through Modbus

Controller + communication emulated by 4DIAC

Signal-based Synchronization between Simulators (TC 3)

OLTC & Remote measurement

- *FMI interface for NS-3*
- *Two models of computation (communication + grid)*



OLTC: On-Load-Tap-Changer

Challenges of FMI-based co-simulation of communication network models

- co-simulation of *physical systems*:
  - exchange of information that corresponds directly to *physical properties* (voltages levels, temperatures, etc.)
  - send *values* of associated model variables from one simulator to another
- *communication systems*:
  - do not just exchange values, but *messages*
  - *transmission* with the help of *protocols* (metadata, data formats)
  - communication network simulators provide *dedicated functionality* to handle the details of data transmission protocols
- challenges regarding FMI
  - provide no functionality regarding message transmission

    → details have to be hidden behind FMI-compliant co-simulation interface of the simulator
  - limited support for event-based co-simulation

    → no support for event detection or event prediction

    → no notion of an input or output being absent

Proposed FMI-compliant approach: Data exchange with message-based simulators

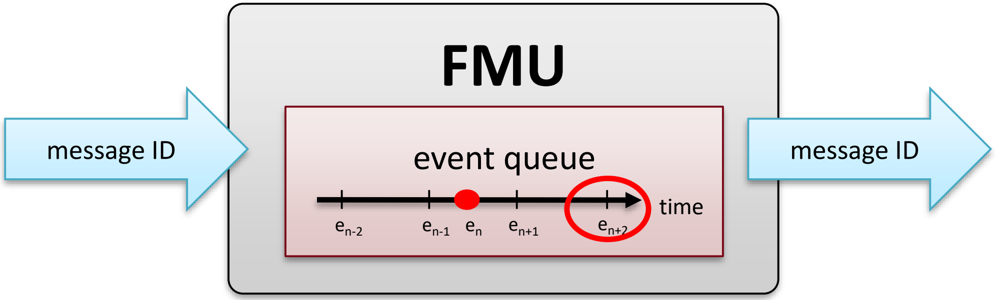
Details of data transmission protocols must be *hidden behind the FMI-compliant interface*:

- *message IDs*
  - transmitted data is associated with a unique message ID
  - message ID is being forwarded to the simulator
- *mock-up messages*
  - simulator generates an internal mock-up message associated with the message ID
  - network model is executed with the mock-up message as stand-in replacement for the original data
  - no need to consider the translation of the original data into a proper format for transmission
  - once the mock-up message has propagated through the network model, its message ID is passed back to the co-simulation framework
- *absence of messages*
  - based on the concept of unique message IDs, a special value represents the absence of input and output messages

Proposed FMI-compliant approach: Event handling for FMUs for Co-Simulation (1/2)

two types of events are of special interest:

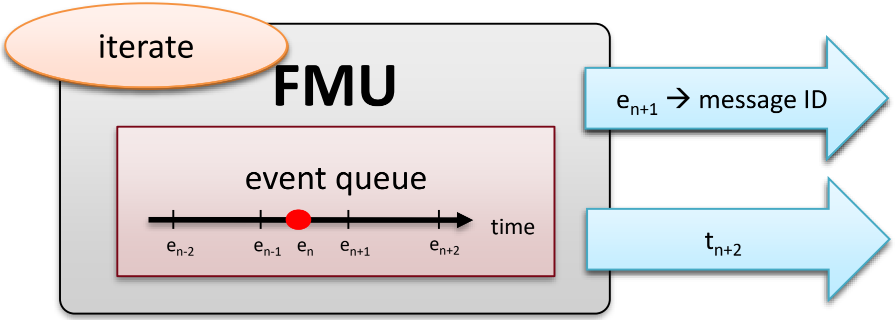
- *input events*
  - mark the arrival of new messages at an input of the simulated communication network
  - value of an associated FMU input variable changes from 0 to the corresponding message ID
- *output events*
  - marks the arrival of a message at an end node in the communication network simulator
  - corresponding output message ID as the value of an associated FMU output variable



FMU: Functional Mock-up Unit

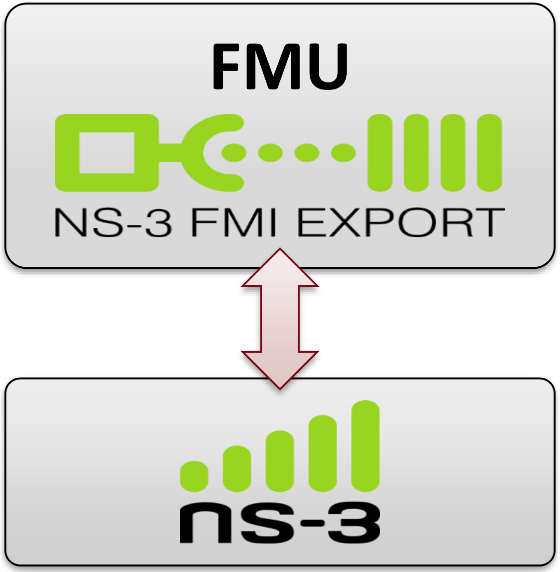
Proposed FMI-compliant approach: Event handling for FMUs for Co-Simulation (2/2)

- FMI specification does not (yet) support the handling of (internal) events for FMUs for Co-Simulation
- ”quick-and-dirty” solution → demonstrate feasibility of approach, but do not put too much focus on specific proposal for FMI extension
  - *internal event prediction*
    - FMUs have to define a dedicated output variable for event prediction
    - value always corresponds to the time of the next internal event
  - *event processing*
    - use *iterations* (simulation steps with step size equal to zero) to trigger the FMU to process events



FMI-support for the ns-3 network simulator

- ns-3 module *fmi-export*
  - creates an FMU for Co-Simulation from a user-defined ns-3 script
  - implements a tool coupling mechanism
    - control the execution of the ns-3 simulator
    - establish a connection for data exchange during run-time
  - interaction with ns-3 is limited to the repeated execution of the same ns-3 script
    - call the FMU's step method → ns-3 executes the same model
    - use different random seeds each time → produce different outputs
- user has to implement a dedicated class → class *SimpleEventQueueFMUBase*
  - provides functions for declaring input and output variables
  - provides functions for adding events to internal event queue
- open source, available at https://erigrid.github.io/ns3-fmi-export/



Main achievements

- Major steps towards standardised co-simulation
- Further development of FMI++ library and its Python wrapper
- **FMI++ adapters** for
  - Powerfactory
  - ns-3
  - PSCAD
  - Matlab
- **Scalability** of co-simulations demonstrated using
  - Holistic testing approach
  - Realistic cyber-physical energy system test cases

Chapter 3 of European guide to power system testing

Conclusions and lessons learned

- Co-simulation aids in assessing smart grid behaviour in a multi-disciplinary setting
- JRA2 showcased the capabilities of standardised co-simulation for smart grids:
  - Functional mock-up interface
  - Mosaik
  - FMI++
- FMI adapters developed for PowerFactory, Matlab, ns-3
- Programming skills needed running co-simulations as in ERIGrid: applicability shall be improved by automation and FMU coupling as a service.