Introducing co-simulation concept and platforms

Arjen van der Meer

Delft University of Technology

Joint Cineldi/ERIGrid workshop, Trondheim, 2019-10-28

Agenda

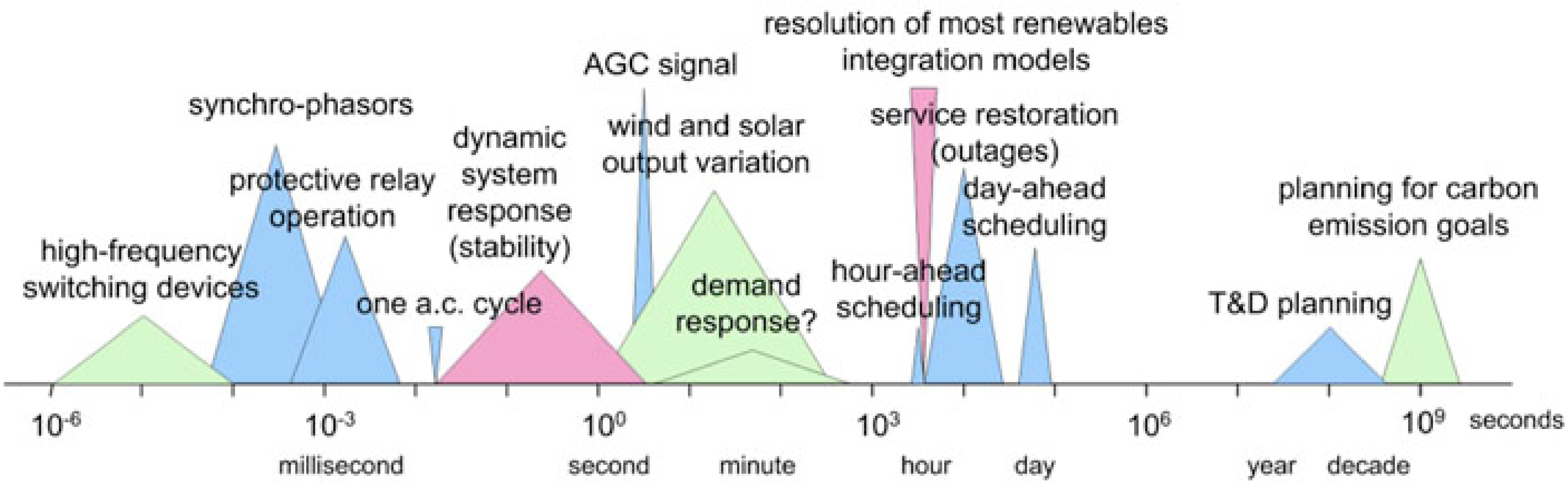
- Cyber-physical energy systems
- Testing and validation approaches
- Introduction to co-simulation
- 2 Implementation Examples
- Platforms and/or frameworks
- aaS

What is going on in the energy system realm?

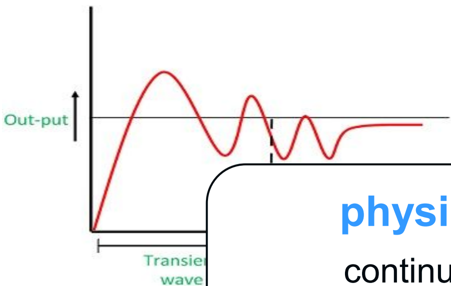
Very multi

- Multi-time scale
- Multi-energy
- Multi-commodity
- Multi-scale
- Multi-domain

Multi-Time Scales



Multi-domain: cyber-physical energy systems



physical world
continuous models

energy generation, transport, distribution, consumption, etc.

roles/behavior
game theory models

agents acting on behalf of a customer, market players, etc.

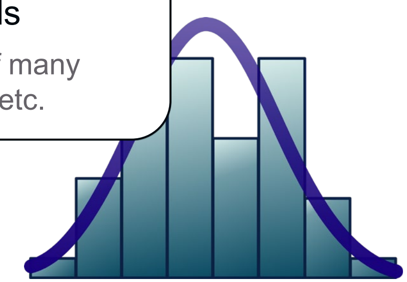
cyber-physical energy system

information technology
discrete models

controllers, communication infrastructure, software, etc.

aggregate/stochastic
statistical models

weather, macro-view of many individual elements, etc.



Holistic testing and validation of CPES

CPES: cyber-physical energy systems

Why ‘holistic’ testing and validation

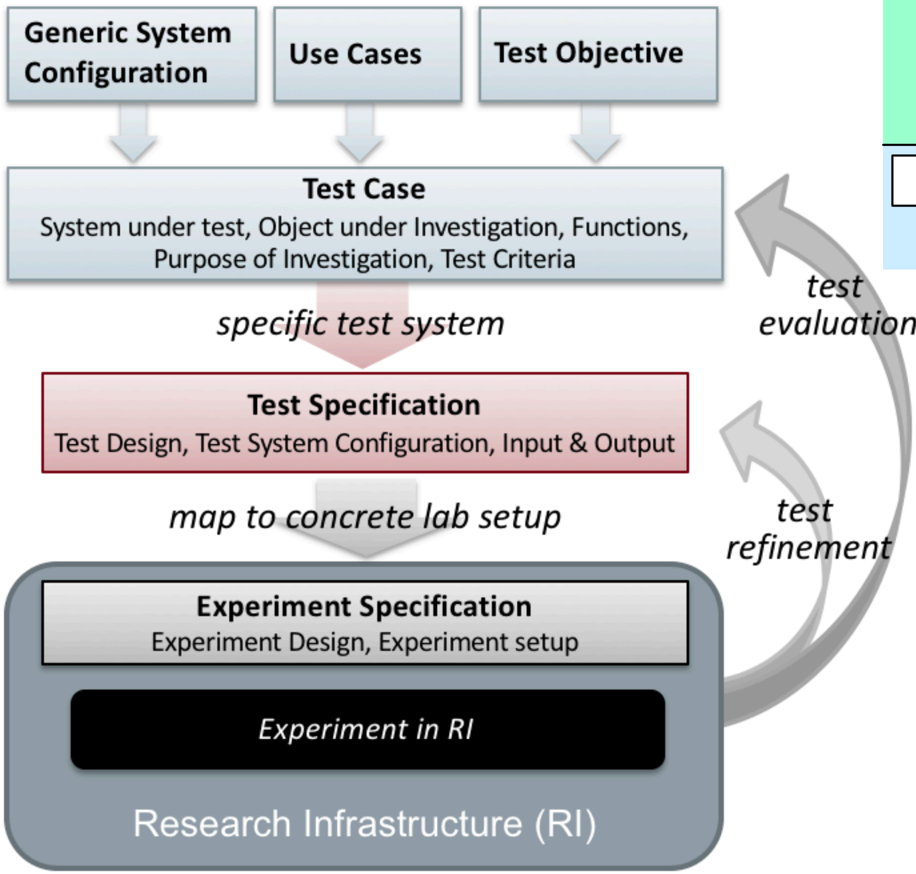
- **Classical** way: split complex systems into subsystems that can be formally validated
- **risk**: cross-domain interactions can develop undesired system-level behaviour
- CPES commonly too **complex** (heterogeneous) to be formally validated
- Experiments, simulations, and Hardware-in-the-loop form an integral part of the validation procedure
- Involves labs with different background and speciality: testing needs formal description method, harmonised vocabulary, etc.

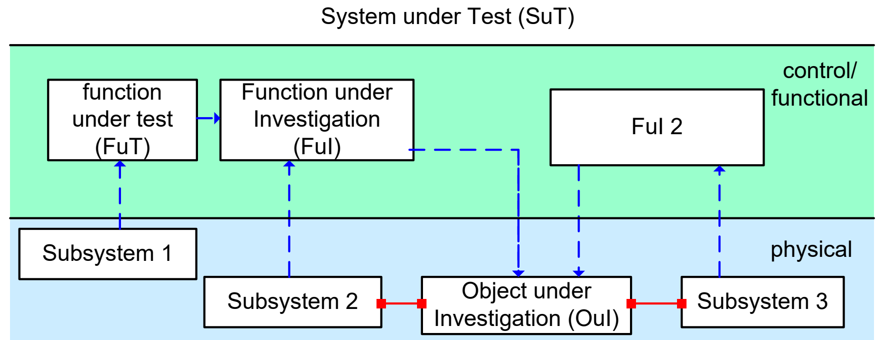
Requirements for holistic testing and validation

- specification must account for system-wide and multi-domain aspects
- specification should allow **multiple** and **diverse** experiments to assess test criteria
- experiment implementation independent from test specification
- build on approaches used earlier (SGAM, UML, SysML, CIM)
- experiments must be **comparable** and **reproducable**

SGAM: smart grid architecture model
UML: Unified modeling laguage
SystML: System modelling language
CIM: Common Information Model

The start: unified test descriptions



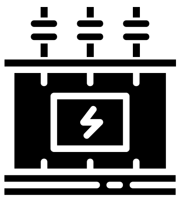


- test case: general system and function specification (**why&what**)
- Test specification: **how** to test the test case
- Experiment specification: how to implement the test specification inside a particular lab

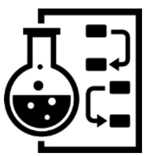
"ERIGrid Holistic Test Description for Validating Cyber-Physical Energy Systems", *Energies*, 2019, https://doi.org/10.3390/en12142722

Simulation-based approaches

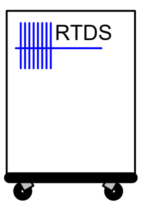
$\sqrt{\frac{x}{y}}$ Analysis → too complex



Real-scale component testing → depends on context



Laboratory testing → System under Test limited size



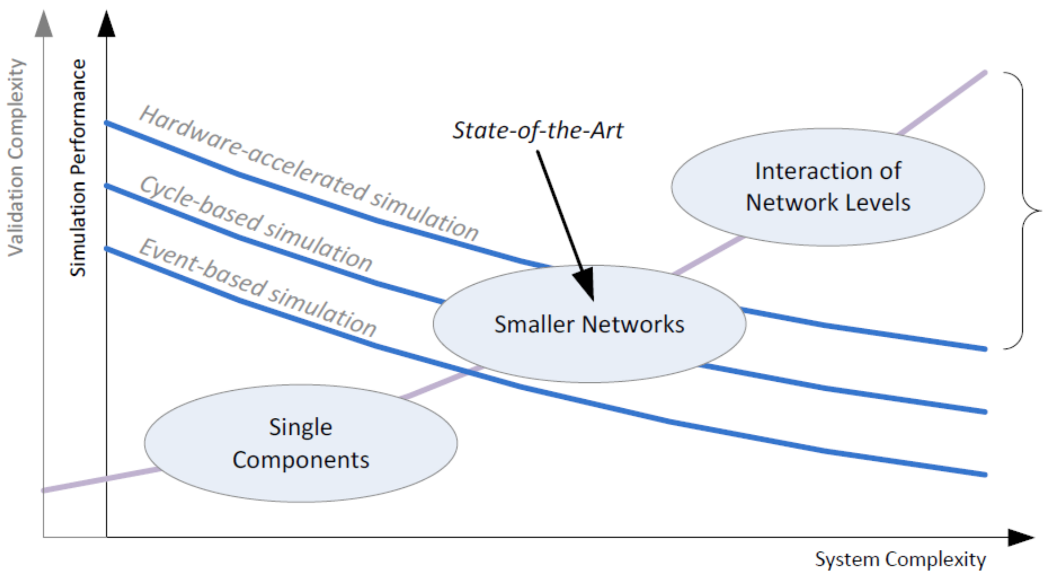
Hardware-in-the-loop testing → later this workshop

Desktop simulation → all virtual

EU H2020 Programme GA No. 654113

The simulation challenge

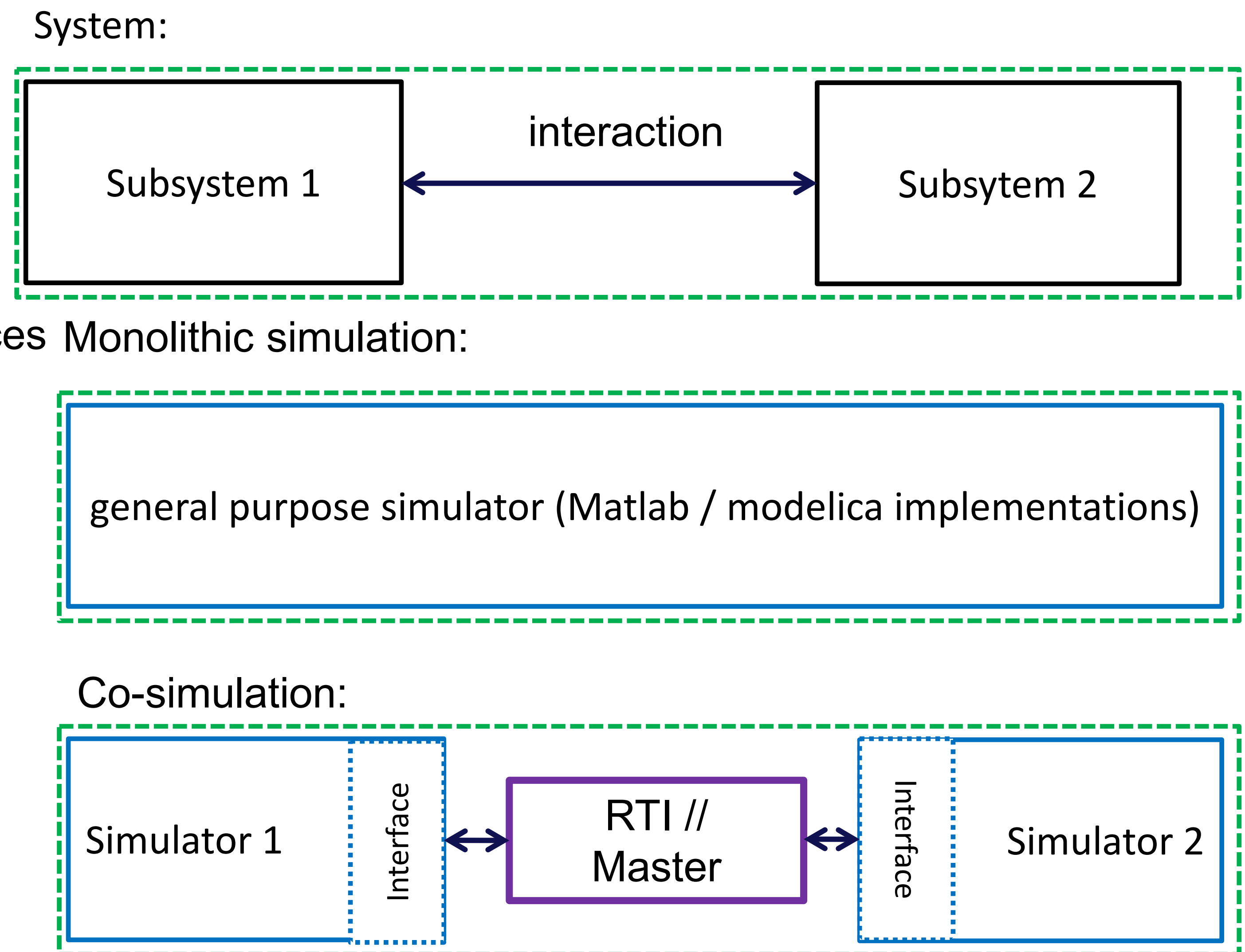
- Simulation forms an integral part of **testing and validation** procedure
- State-of-the-art:
  - domain-specific simulation: fast, but validity and accuracy limited
  - General-purpose simulation: easy prototyping, accurate but scales badly



More Iron?
model aggregation?
Hybrid modelling?

EU H2020 Programme GA No. 654113

“The" solution: co-simulations



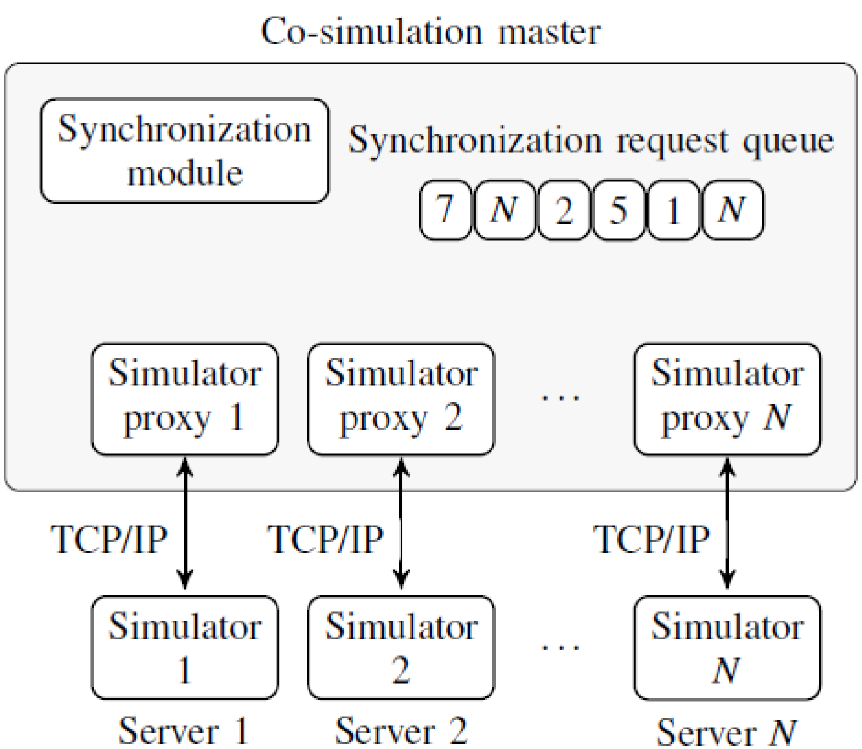
Use specialised tools
Standardised interfaces
Good accuracy
Good performance

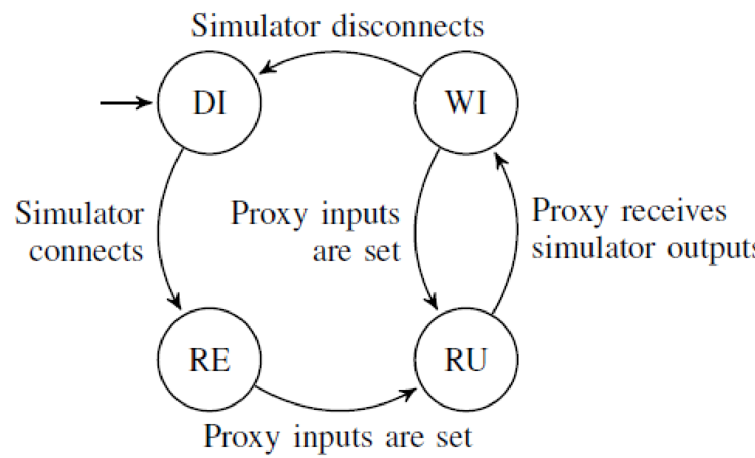
Implementation
Scalability
aaS

RTI: runtime environment

Master algorithm

- A software code which manages synchronization and message exchange

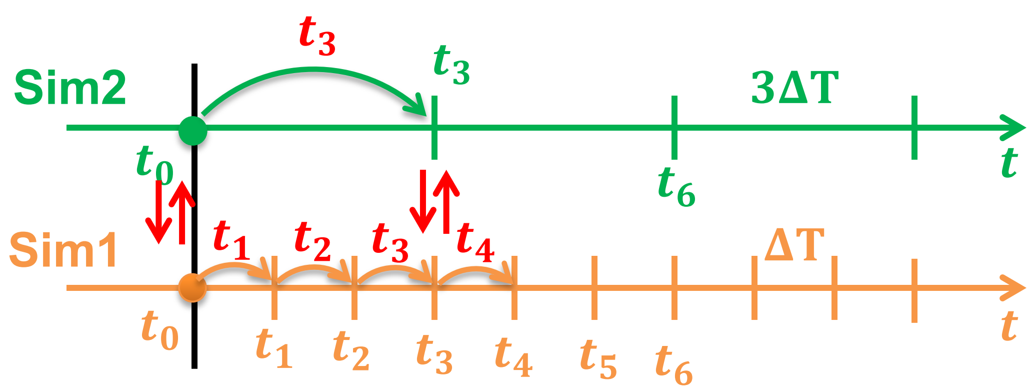




C. D. Lopez, A.A. van der Meer, M. Cvetkovic, P. Palensky, "A Variable Rate Co-simulation Environment for the Dynamic Analysis of Multi-area Power Systems", IEEE Power & Energy Society PowerTech, Manchester, UK, June 2017.

RE – ready for execution; RU – run;
WI – waiting on input; DI – disconnect

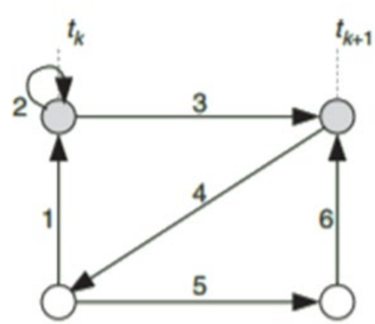
Co-simulation synchronization



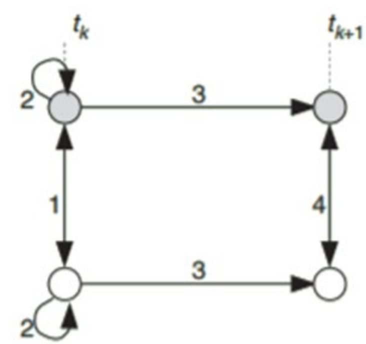
t – global time

t – local time for Sim1

t – local time for Sim2



Serial (Gauss-Seidel)

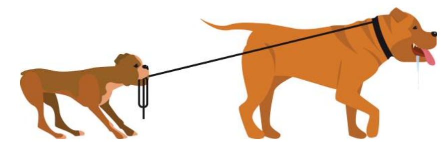


Parallel (Jacobi)

Handling asynchronous events?

Choice of the co-simulation master

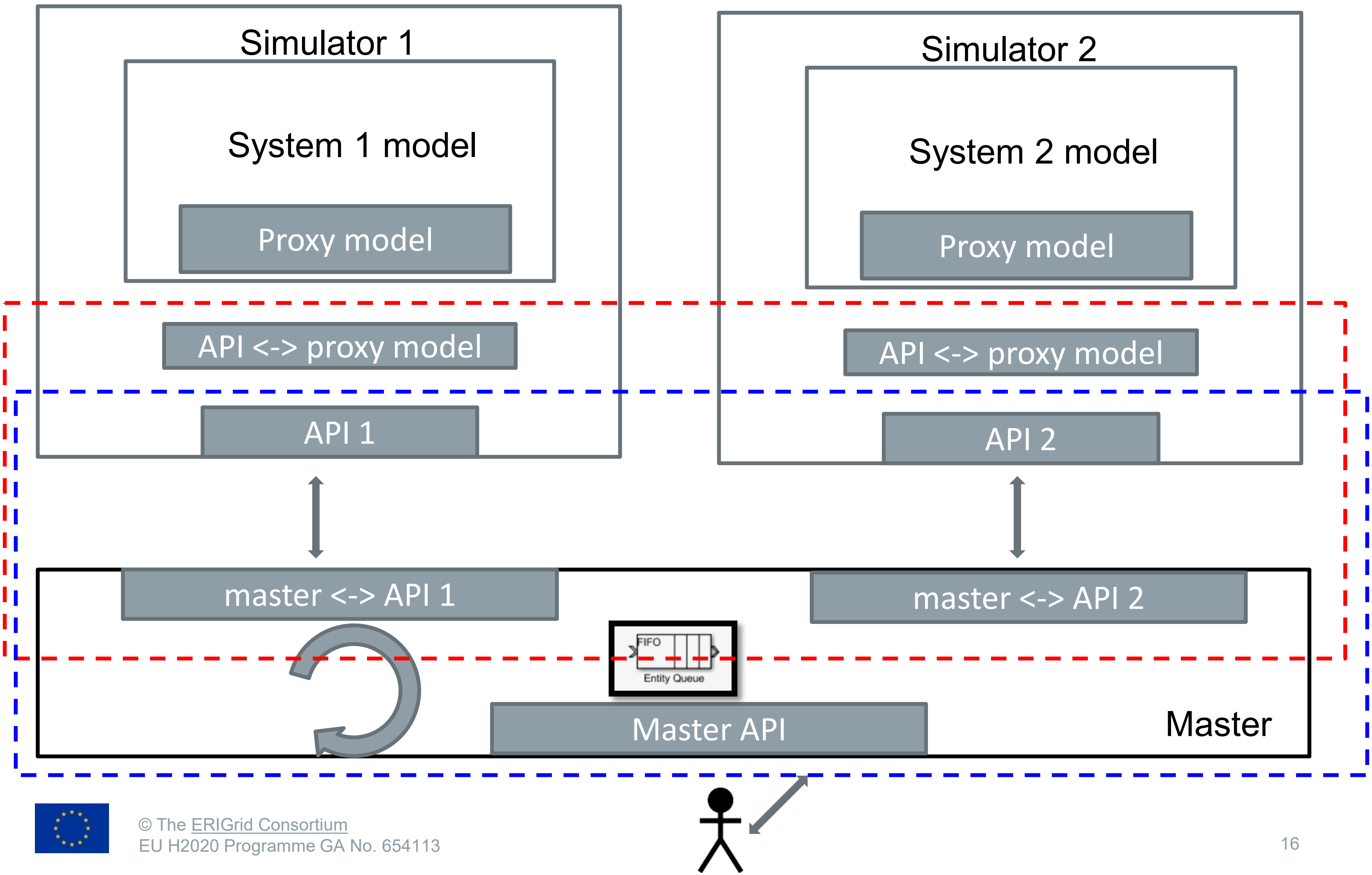
1. One simulator as a master, **example 1 later this presentation**
   - All simulators synchronize to internal clock of one simulator
   - Examples: OMNET++, RTDS
   - Synchronization points and sequence are implicitly defined in this tool



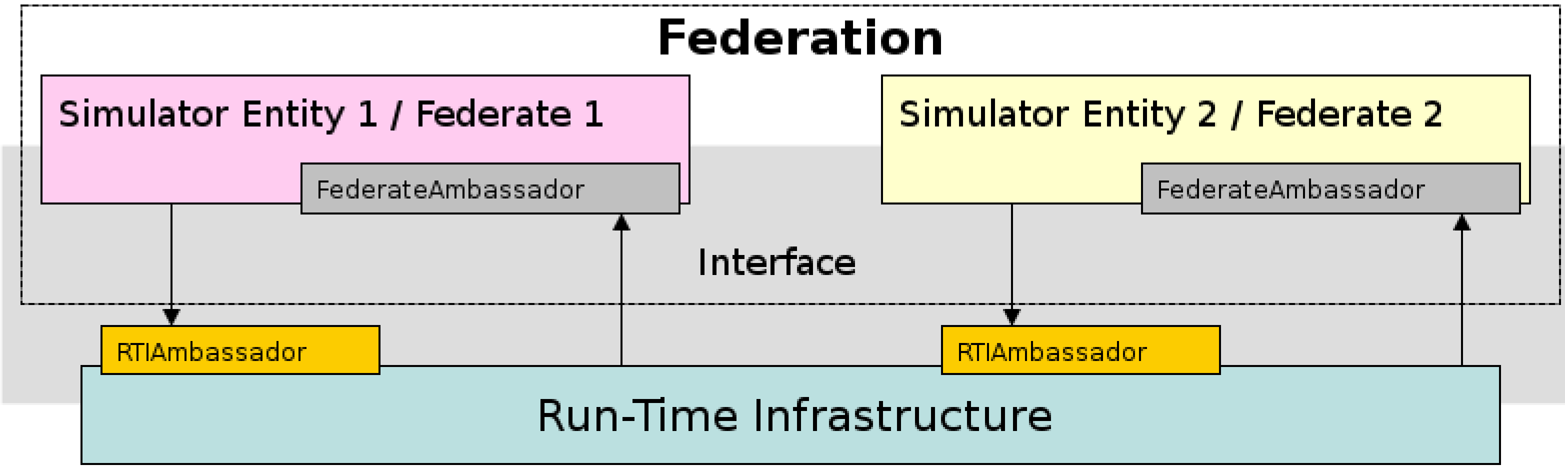
2. Top-down approach (strongly coupled simulators), **example 2 later this presentation**
   - One tool orchestrates the whole co-simulation
   - Example: Mosaik
   - Synchronization points and sequence are explicitly defined in the master code

3. Bottom-up approach (loosely coupled simulators)
   - Each simulator decides on its synchronization method
   - Example: HLA
   - Synchronization points and sequence are explicitly defined by communication points and synchronization requests of each simulator

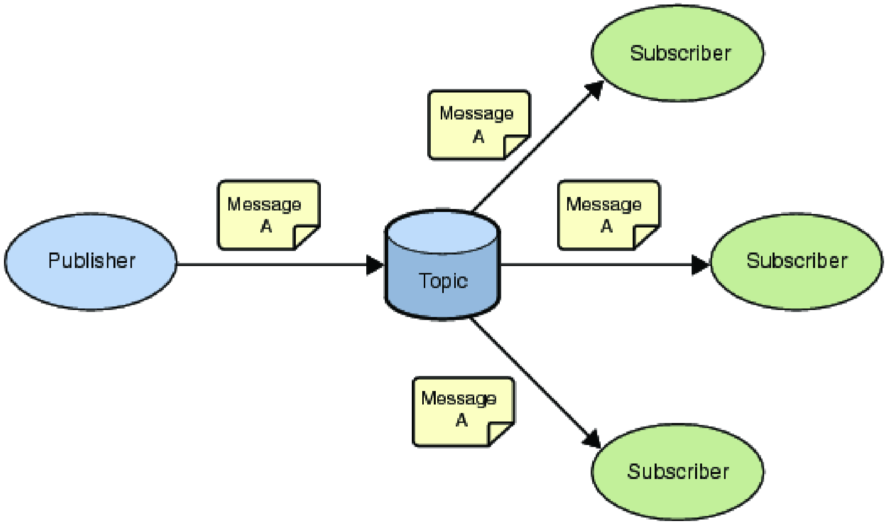
The Simulator interface Hamburger



Master standardisation: HLA, master aaS



- RTI is service provider:
  - Federation management
  - Declaration management
  - Object management
  - Ownership management
  - Time management
  - Data distribution management
  - Support services



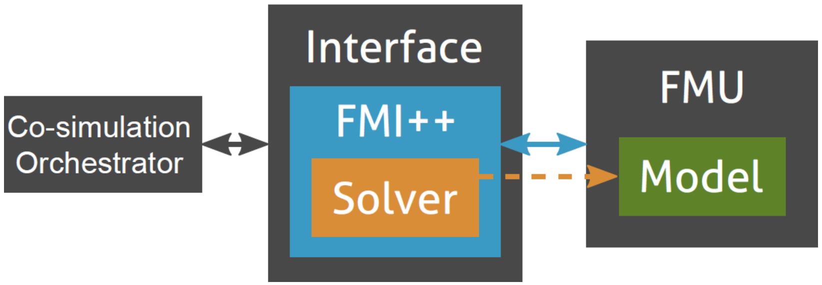
Pros: highly reconfigurable

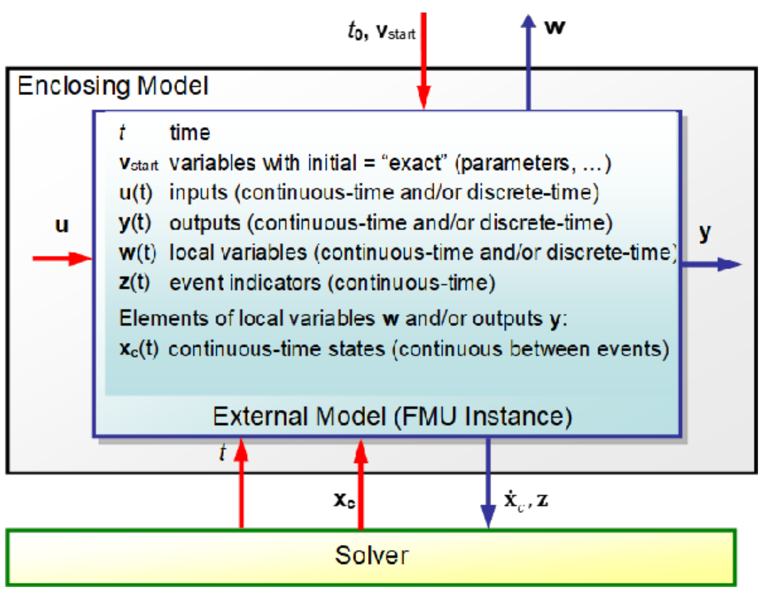
Cons: steep learning curve. Overkill for many applications.

Interface Standardisation: the Functional Mock-up Interface

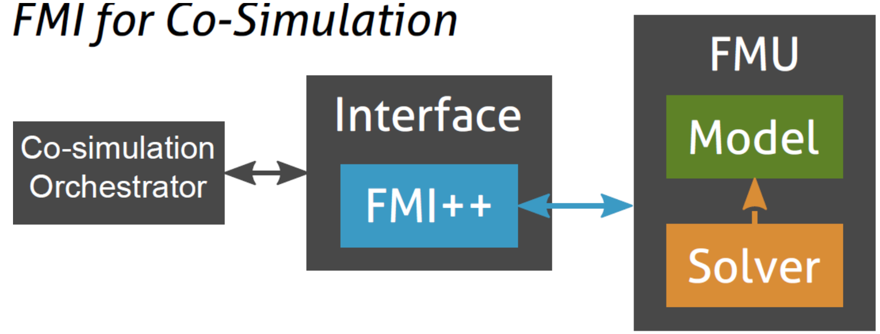
https://fmi-standard.org

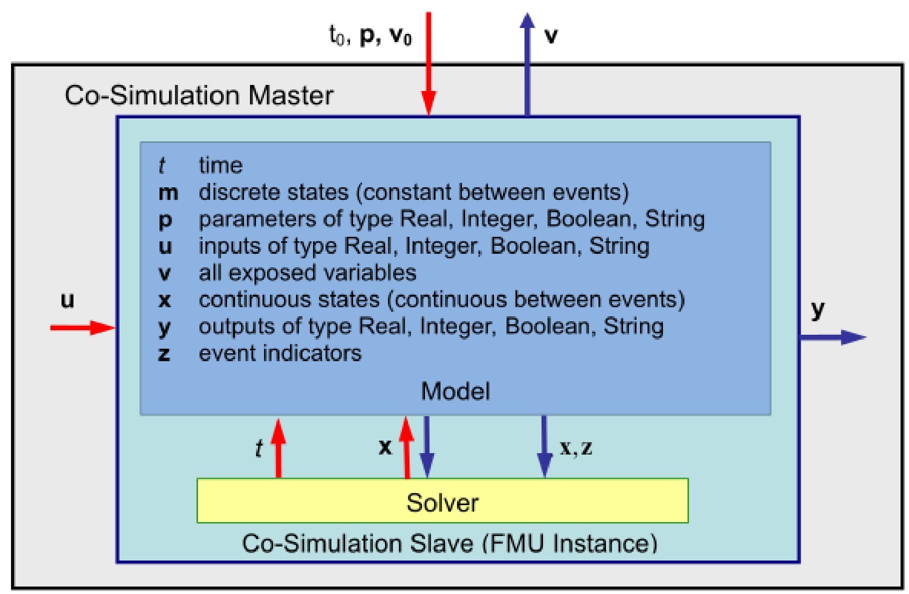
FMI for Model Exchange





FMI for Co-Simulation





EU H2020 Programme GA No. 654113

Implementation examples

1. Co-simulation between transient stability simulator and electromagnetic transients simulator.
   - Stability simulator is the master
2. Co-simulation between PowerFactory and Matlab/Simulink
   - Python / mosaik used as master

Example 1: hybrid RMS/EMT simulation

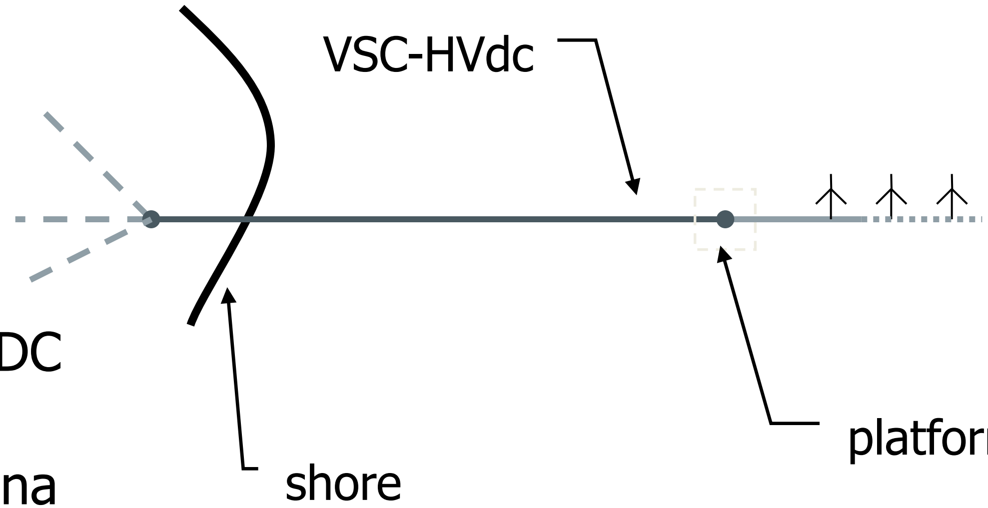
far and large offshore wind

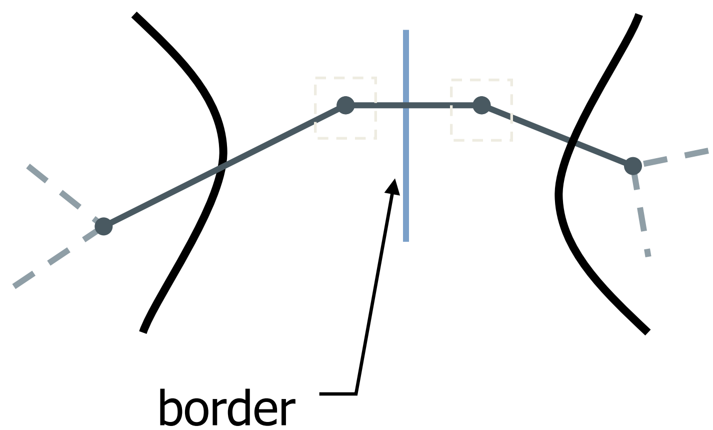
connection through VSC-HVDC

Emergent stability phenomena from HVDC system

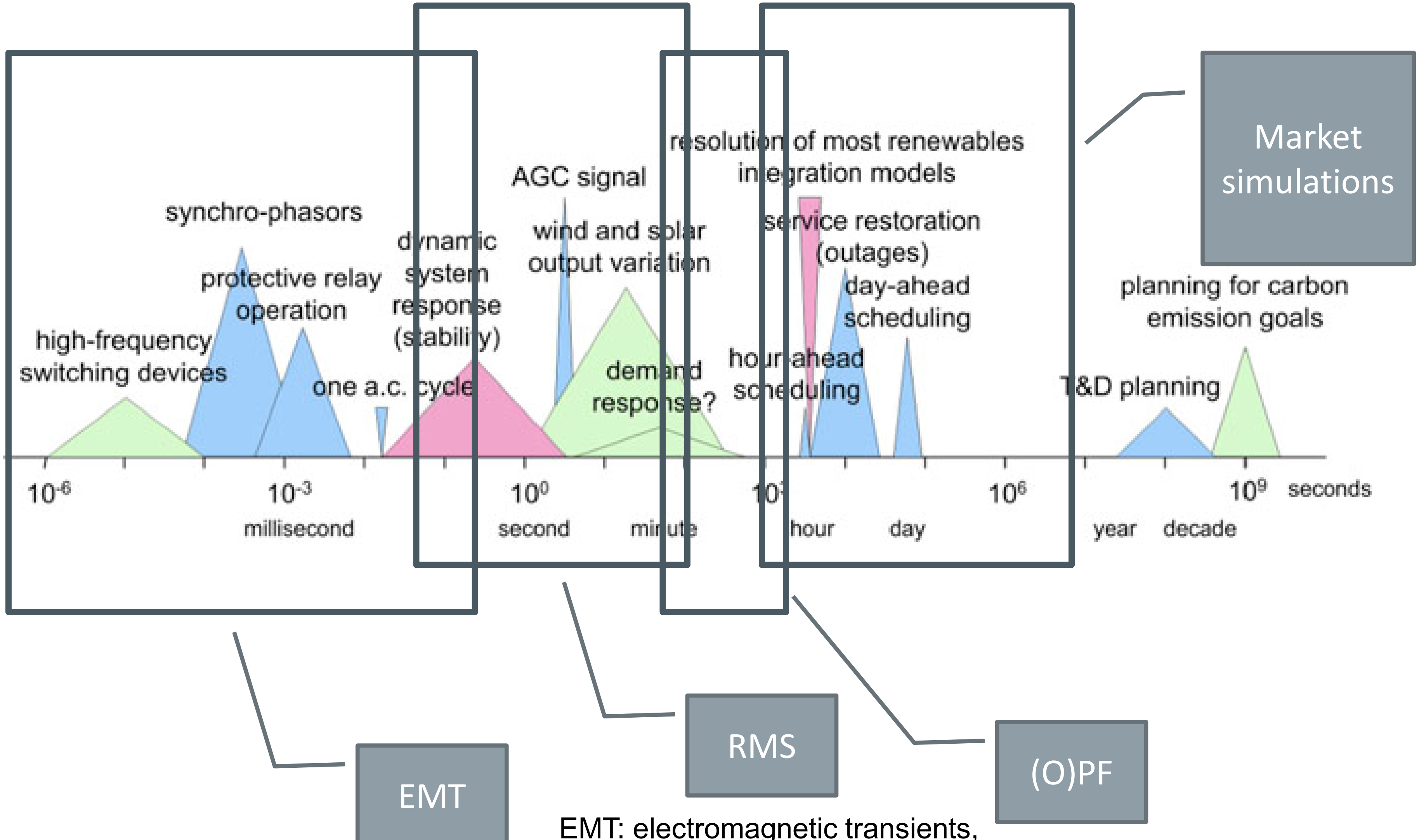
Stability simulations (like PSS/e and PowerFactory) not designed for DC assessment

Model <-> simulator disconnect





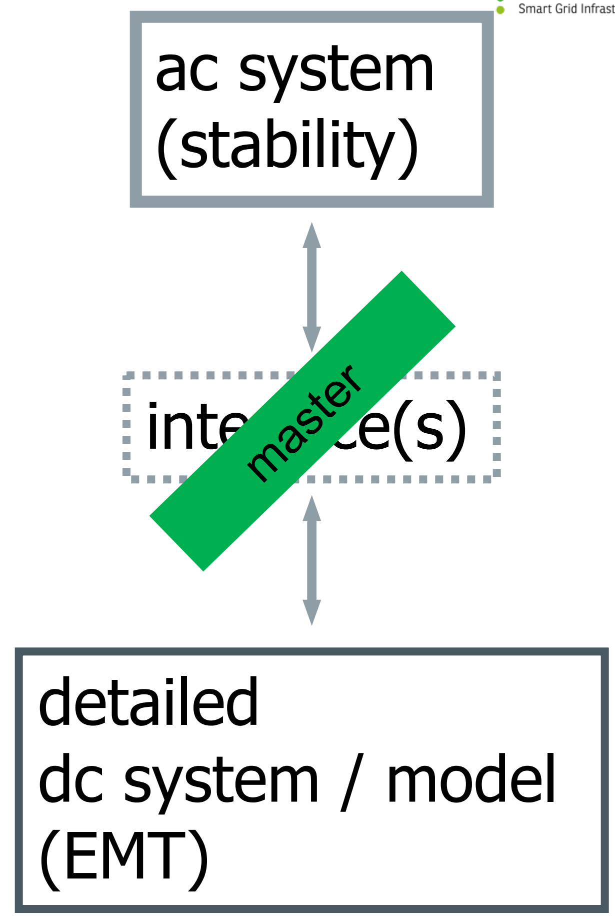
Multi-Time Scales



EMT: electromagnetic transients,
RMS: root-mean square,
OPF: Optimal Power Flow

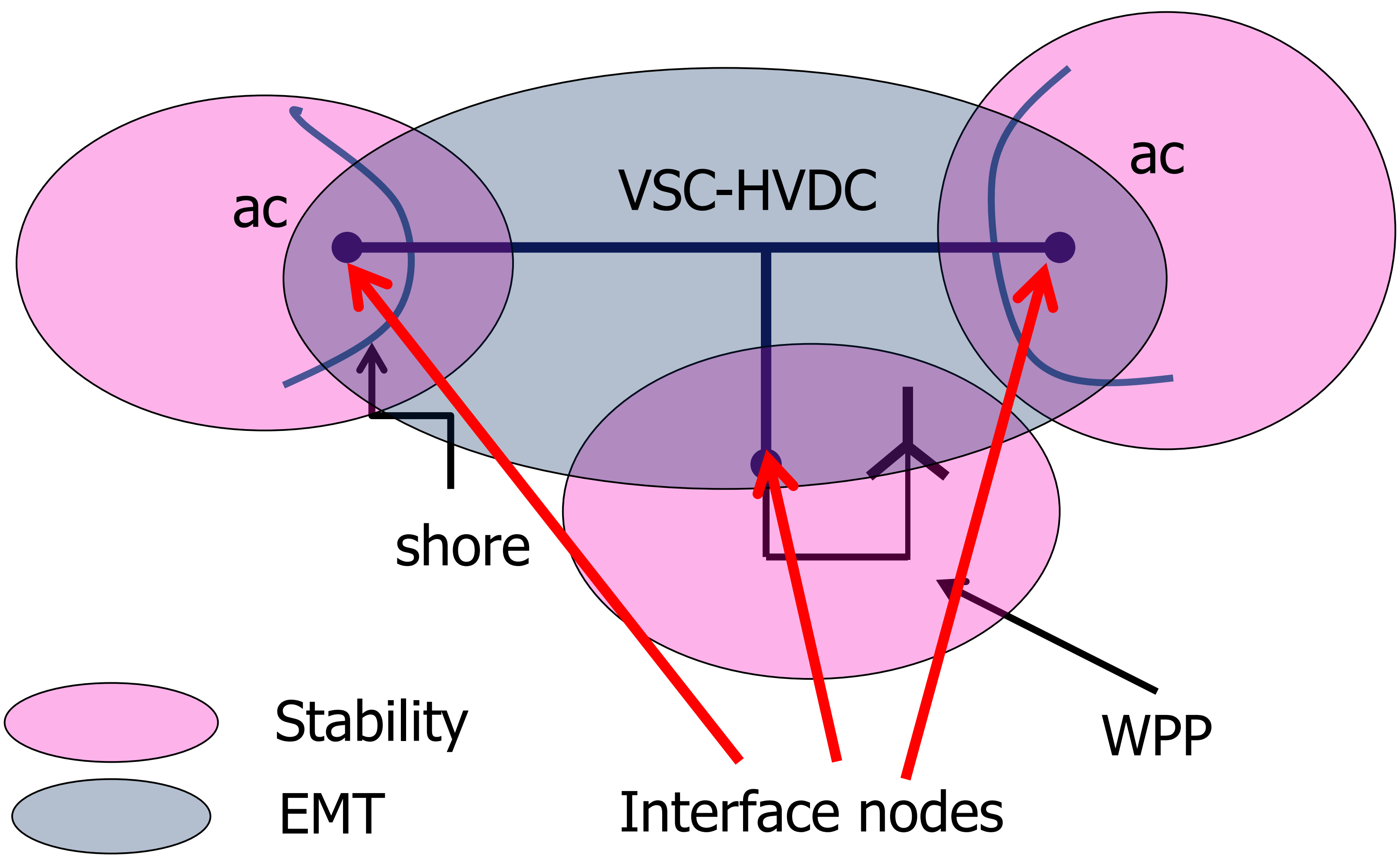
solution: co-simulation

- AC network: stability-type simulation
- VSCs and DC network: EMT-type simulation
- Connection via the interaction protocol (the master algorithm):
  - exchange of variables in pre-defined order



A. A. van der Meer, M. Gibescu, M. A. M. M. van der Meijden, W. L. Kling, and J. A. Ferreira. Advanced hybrid transient stability and EMT simulation for VSC-HVDC systems. *IEEE Trans. Power Del.*, 30(3):1057 -1066, June 2015.

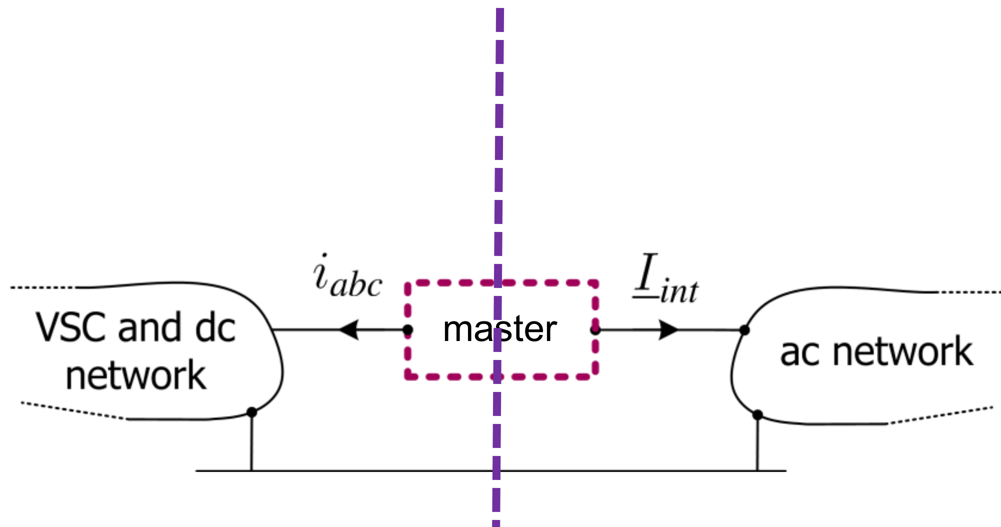
System decomposition



WPP: wind power plant

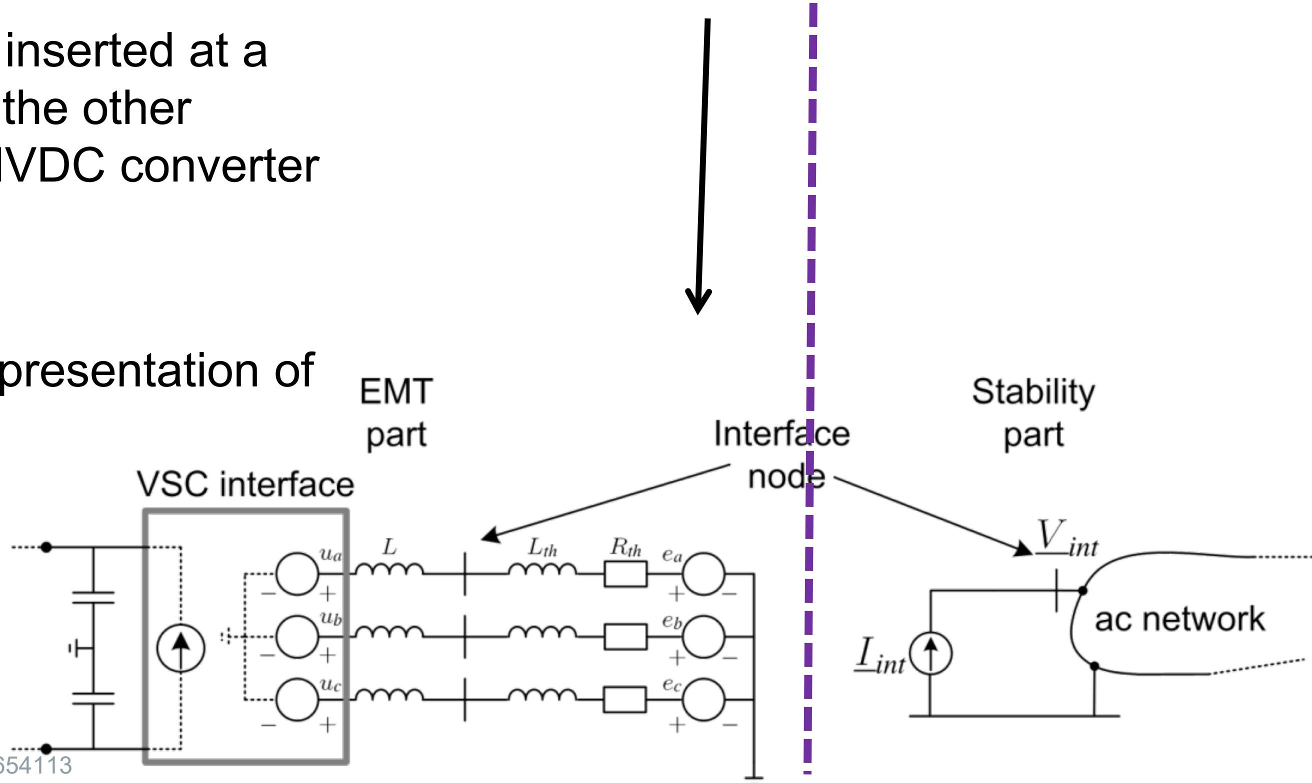
Interface ‘model’

Each simulator needs a **proxy model** (called interface model) of the other simulator



The proxy model is inserted at a physical location in the other network, here the HVDC converter coupling point

Interface model is Thévenin/Norton representation of both systems



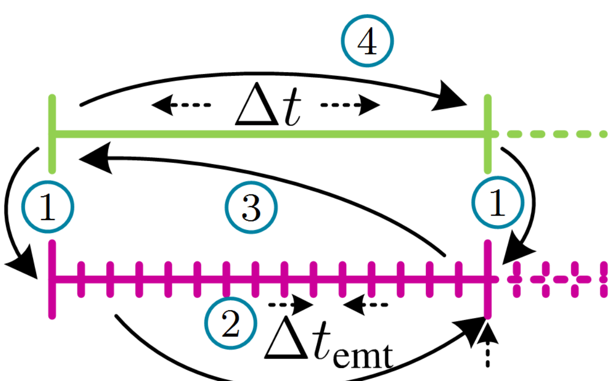
Interaction protocol (**Master algorithm**)

Defines when and in which order the **proxy models** (sources) in both systems are updated

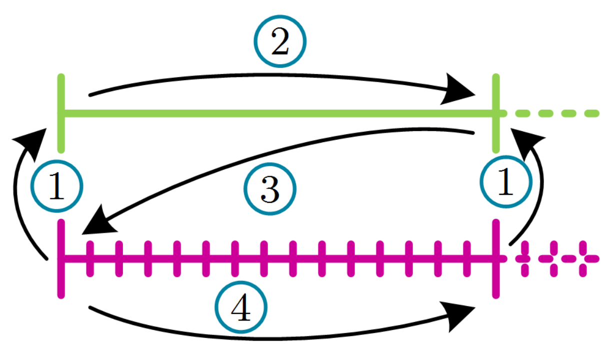
EMT solver is separate (*co-simulation*) or included by a minor integration loop (*hybrid* simulation) → one of the simulator acts as a master

Variable sources updated at fixed instances

EMT priority:

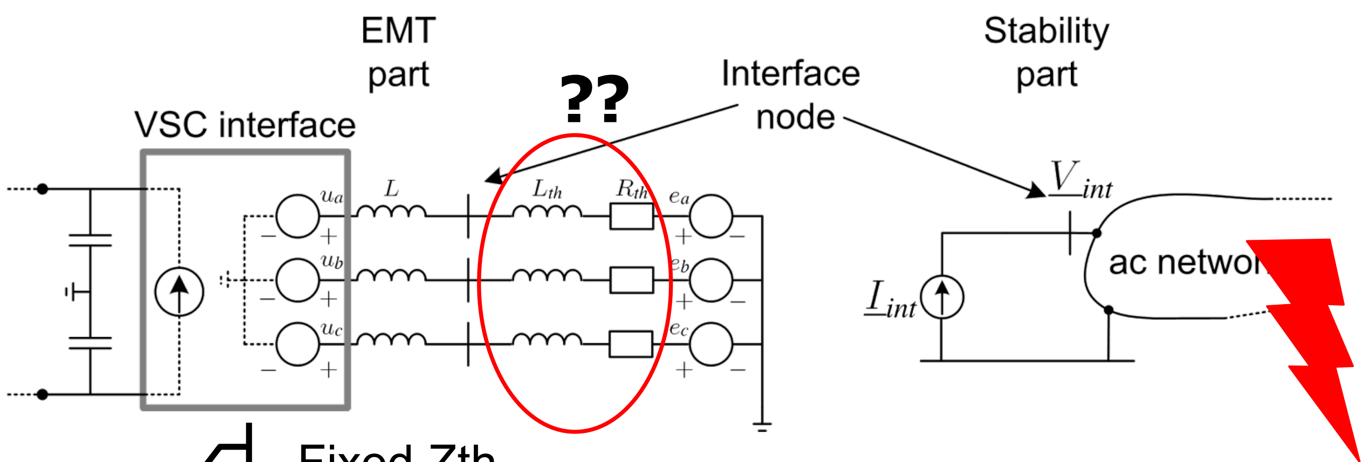


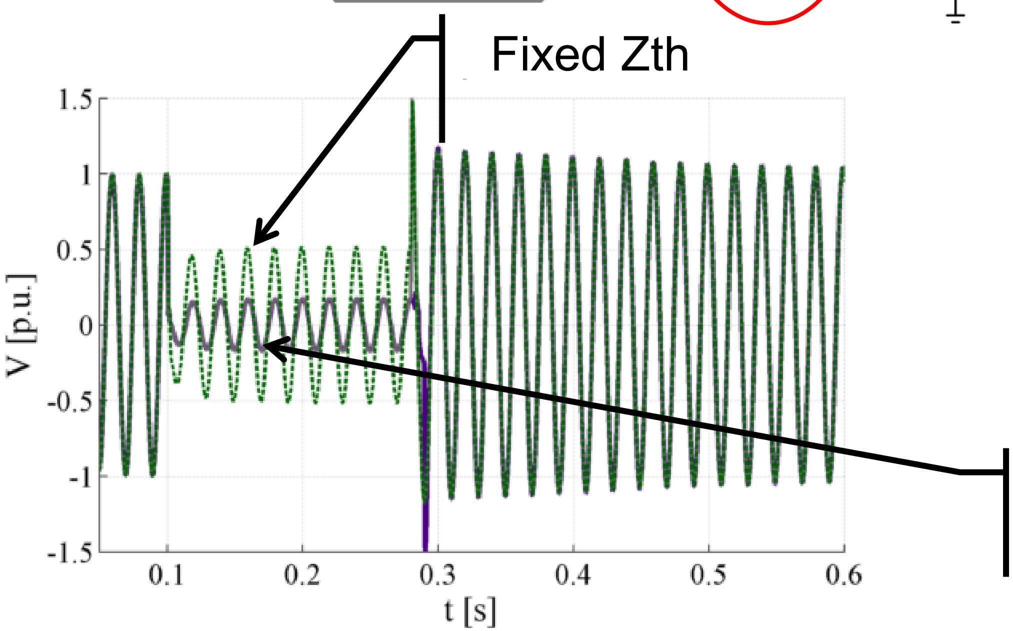
Stability priority:



Remember this one!

Thévenin impedance after faults





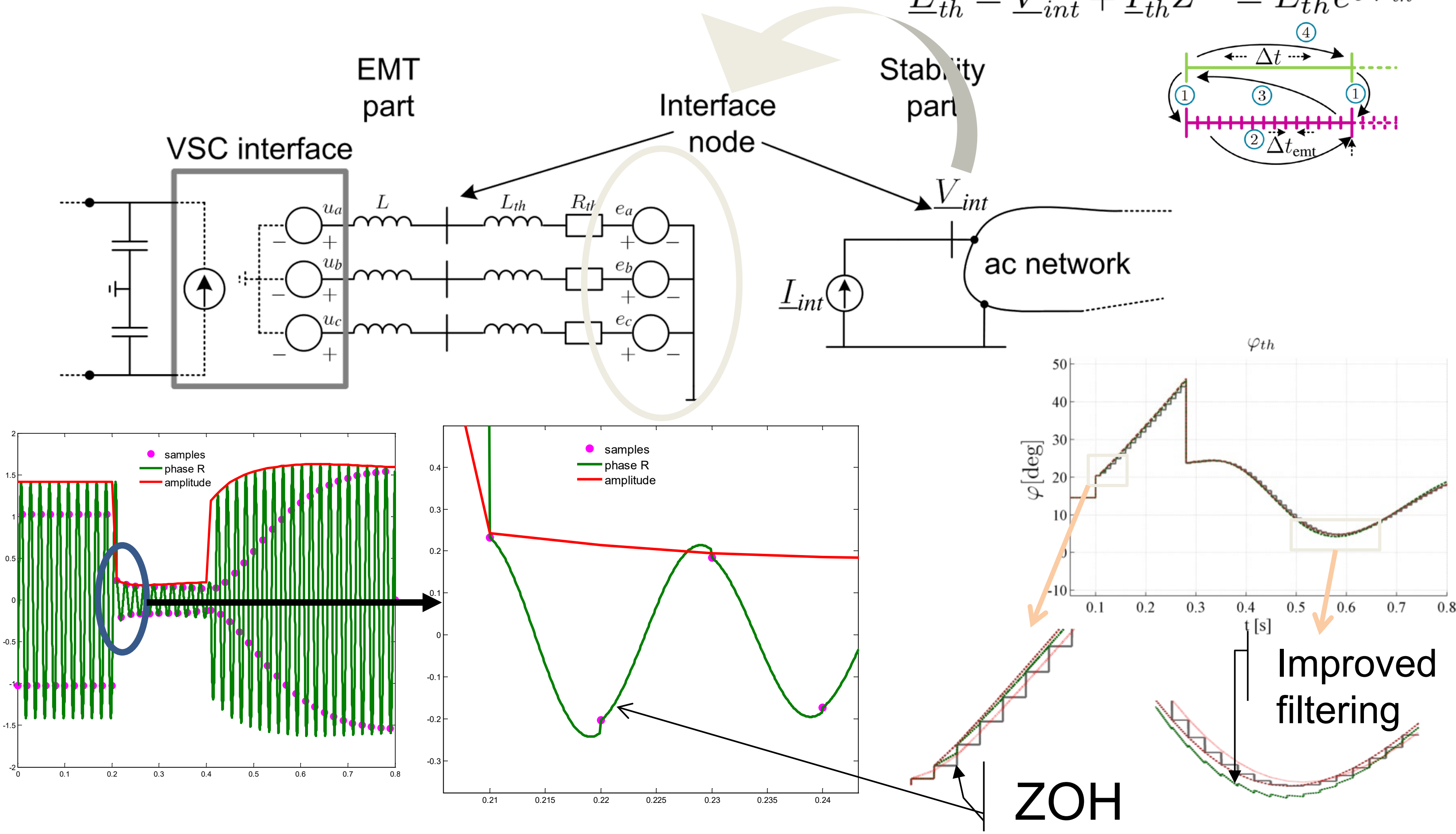
Requires:

- re-initialisation of detailed system
- advanced access to EMT simulation by its API

Improvement & EMT reference

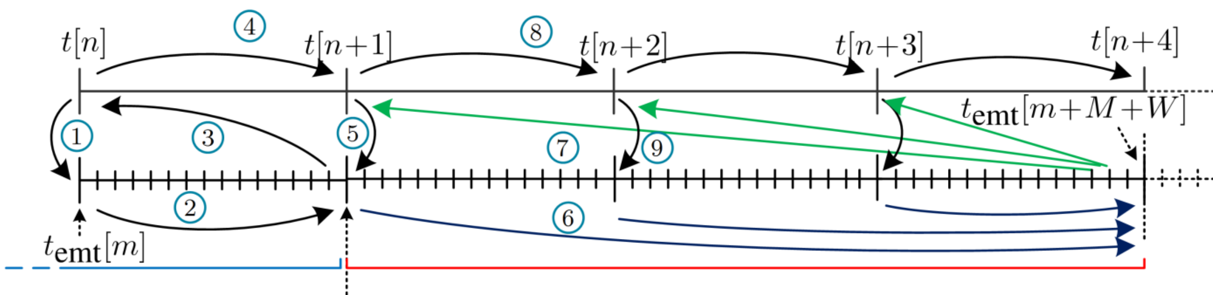
The need for sophisticated proxy models

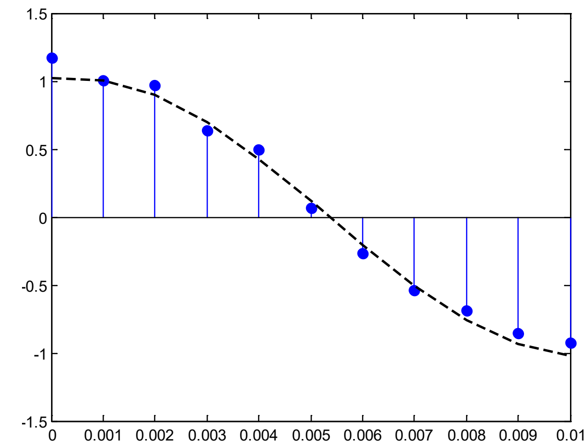
$$\underline{E}_{th} = \underline{V}_{int} + \underline{I}_{th}\underline{Z}^{th} = E_{th}e^{j\varphi_{th}}$$

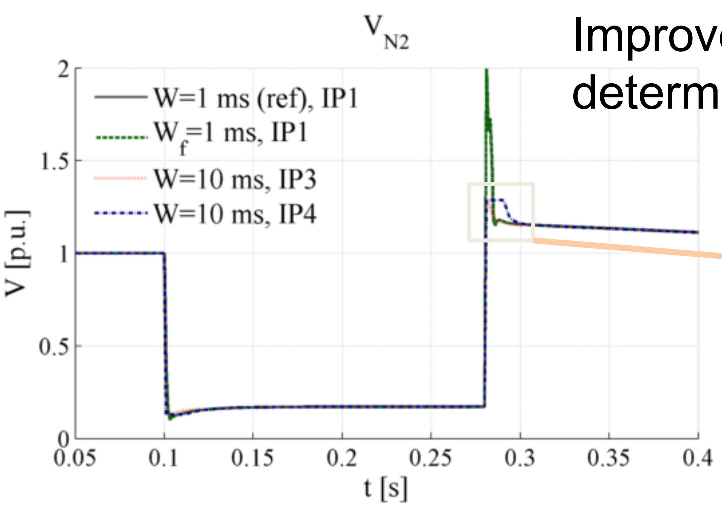


phasor determination if Δt small

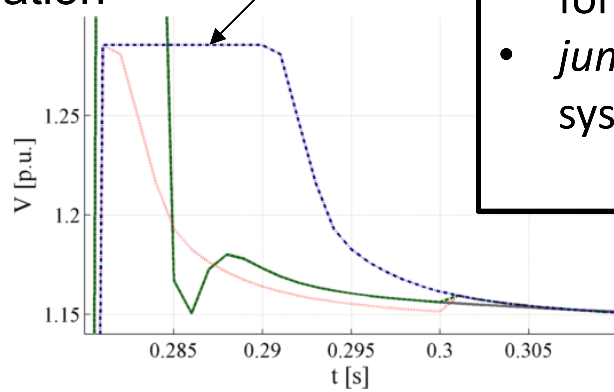
- curve fitting with 10-20 ms window
- Inaccurate after disturbance: procedure needs restart → **asynchronous request**







Improved phasor determination

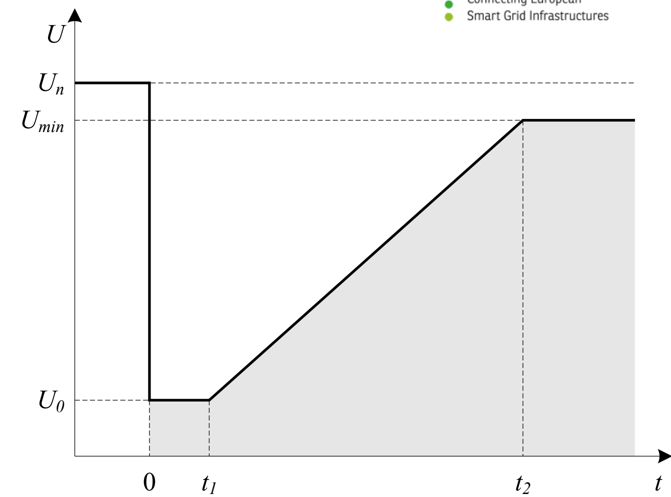


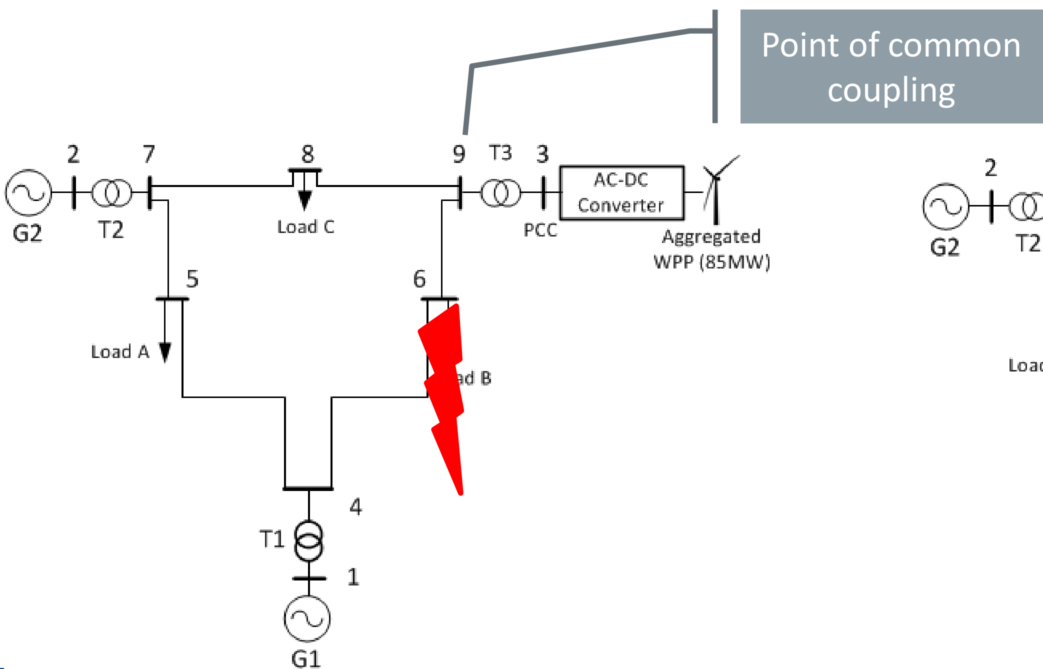
- Accurate curve fitting for small Δt
- *jump-over* in detailed system required

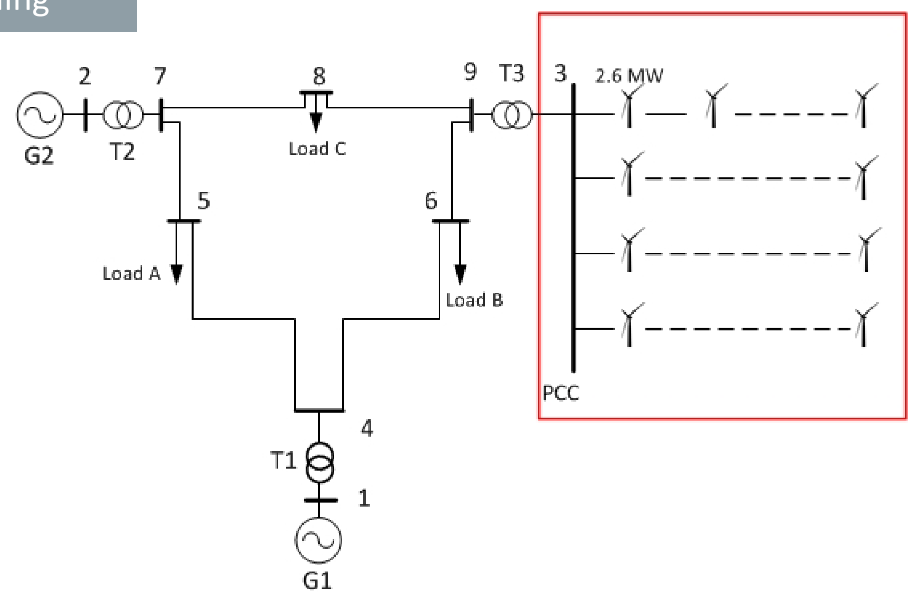
Example 2: PowerFactory / Simulink co-simulation

Case study: fault-ride through of an onshore wind power plant

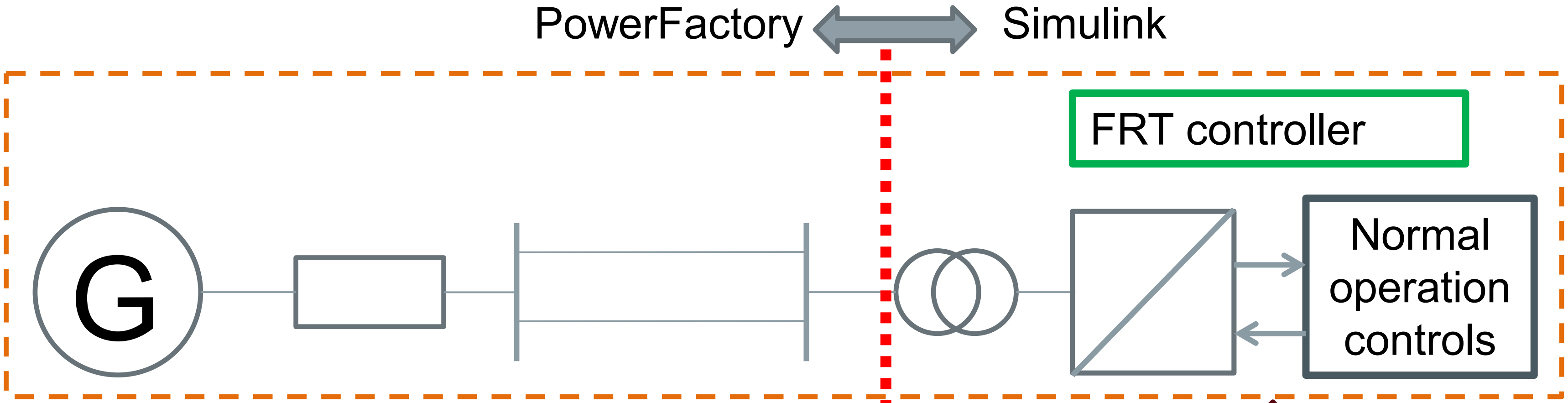
- Fast continuous/discrete system interactions
- Wind turbine model in **Simulink**
- Grid model in **PowerFactory** (RMS-mode)
- How does the simulation scale (accuracy/speed)?

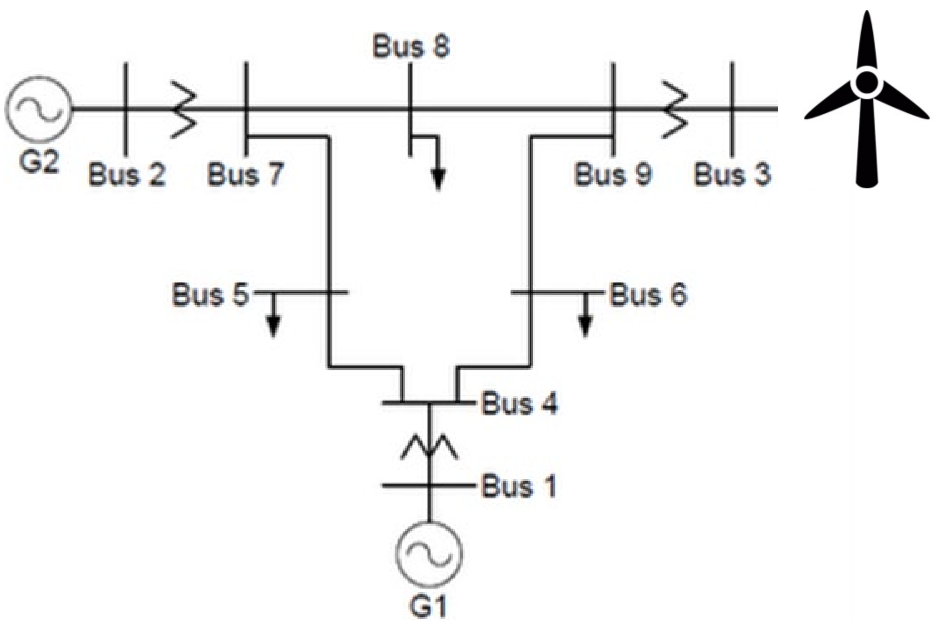


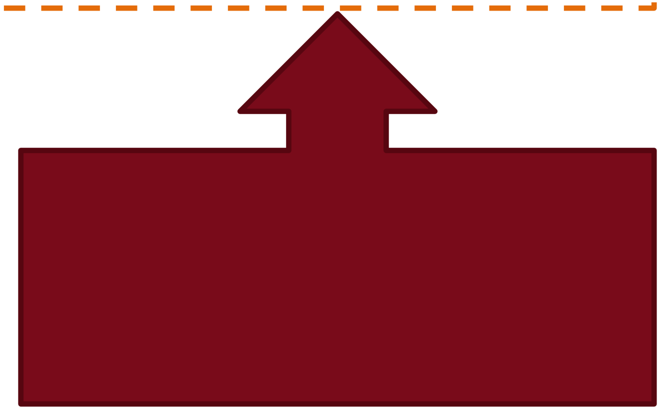




The system under test



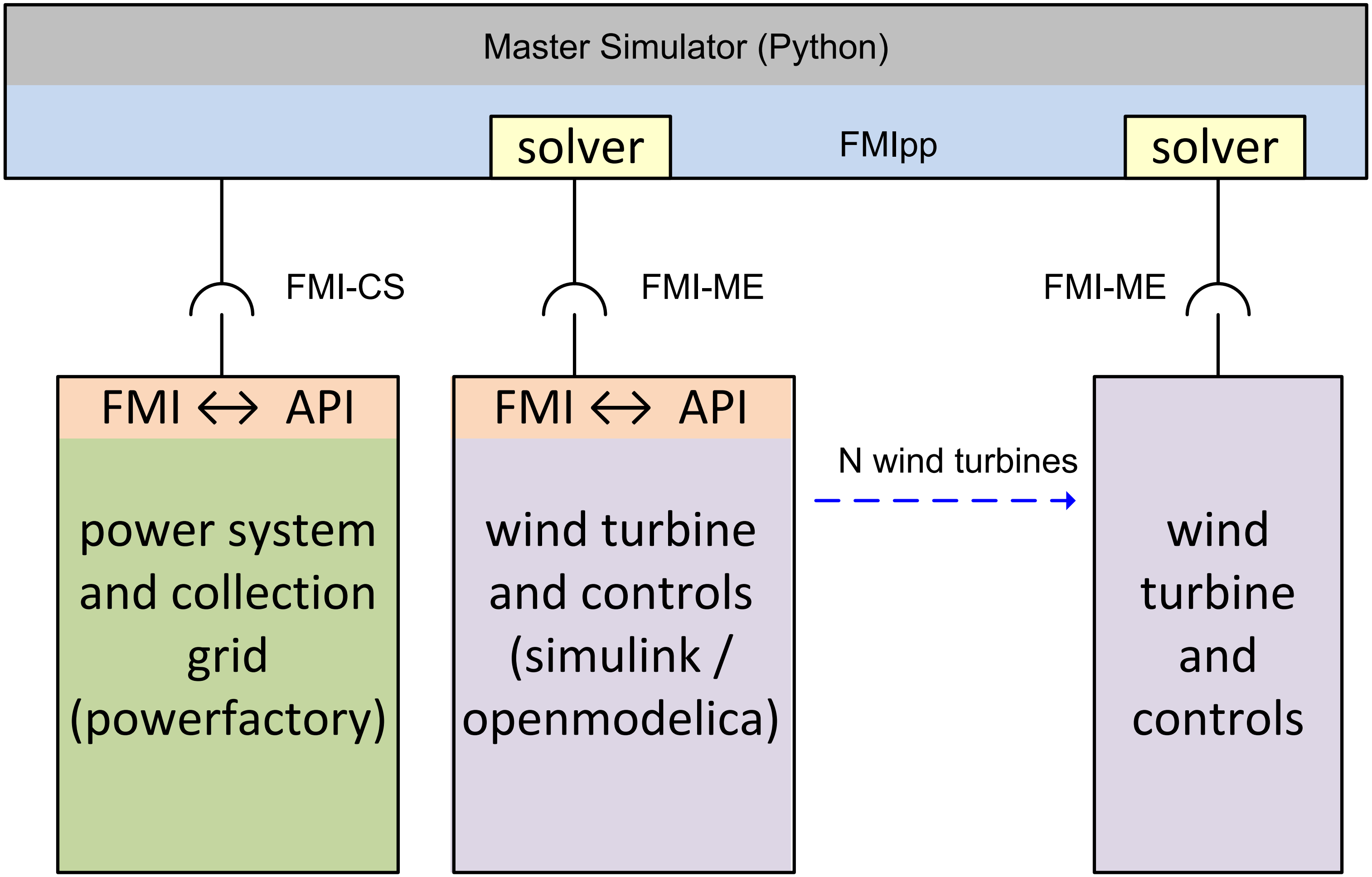




Wind turbine:
FRT and vector controller

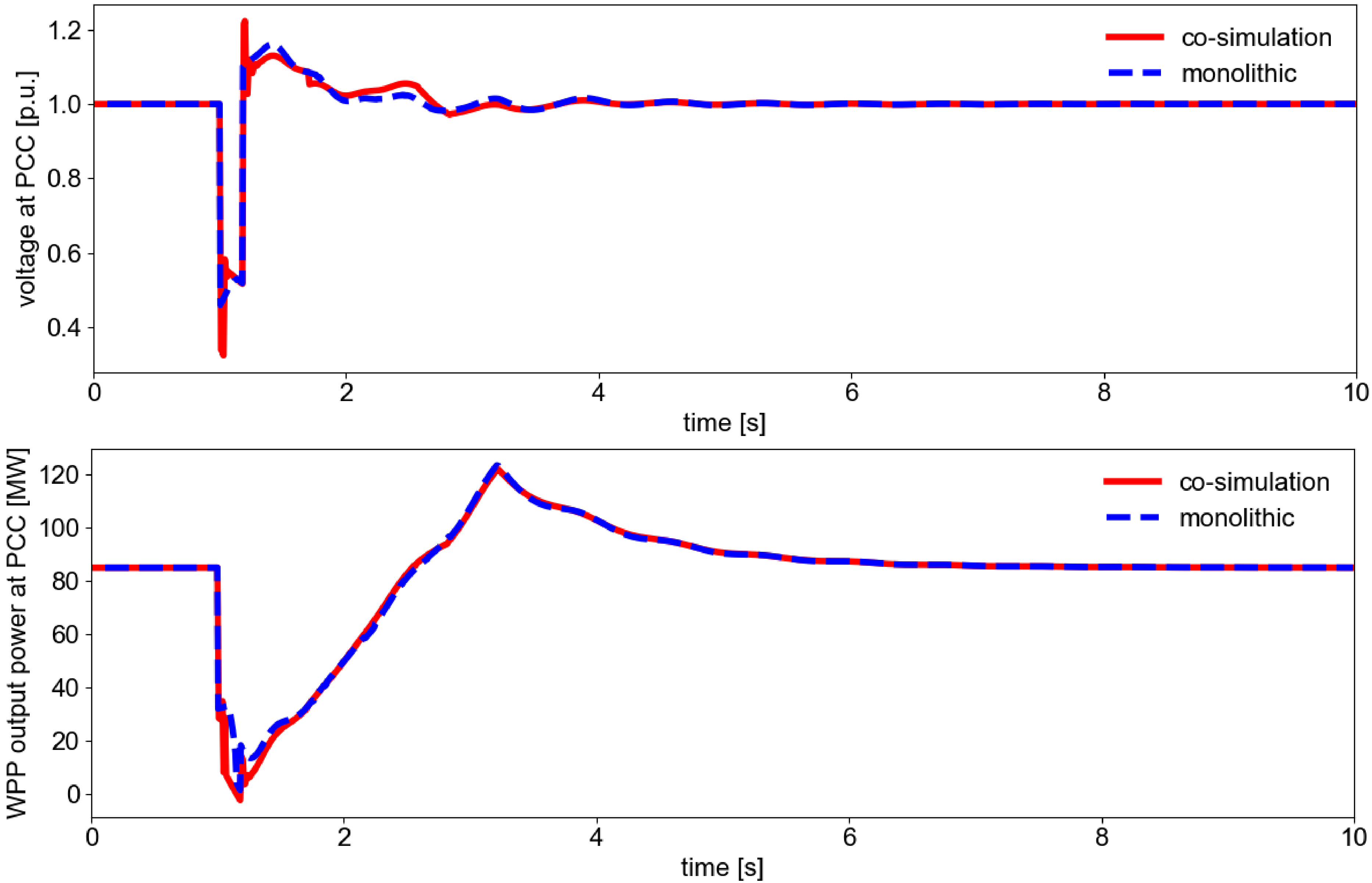
FRT: fault ride-trough

The experiments: setup



EU H2020 Programme GA No. 654113

Simulation results (1)

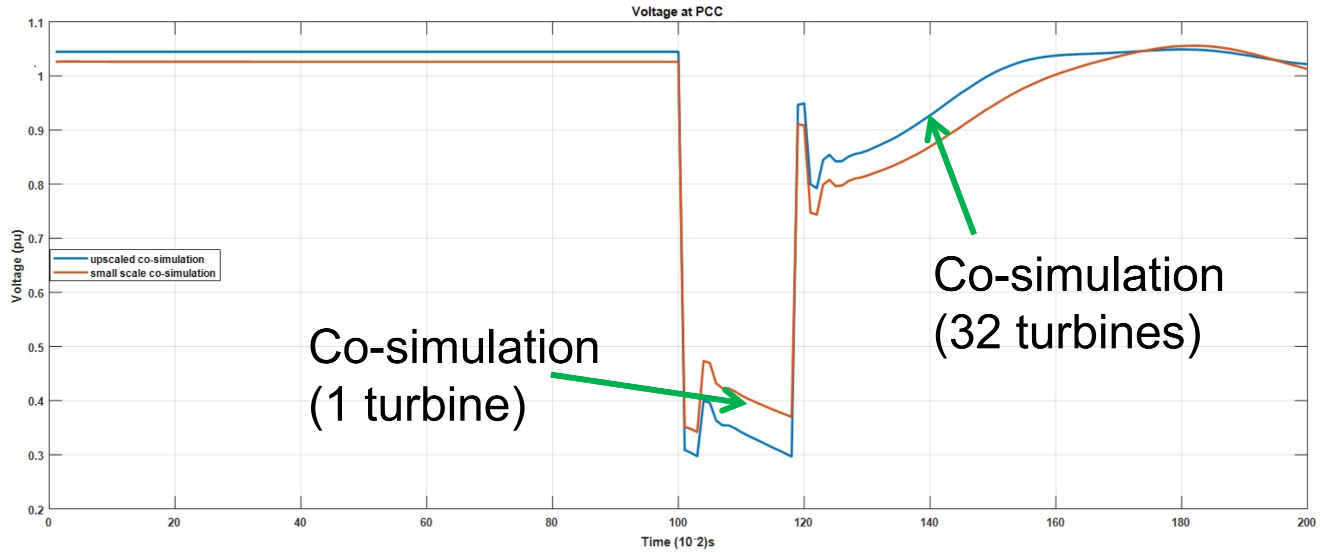


Simulation results (2)

No active power gradient

2 s simulation run

Voltage:



<u>Execution speeds per unit</u>:

Monolythic:

~ *Wall-clock time (1s)*

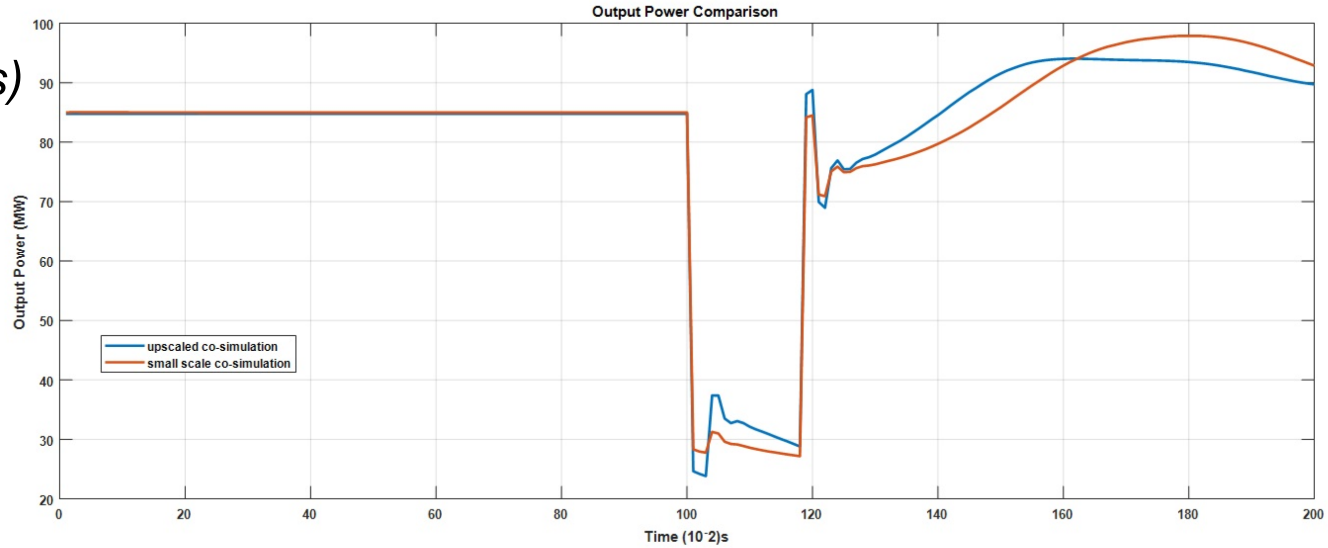
Co-simulation:

(81 s)

Upscaled co-simulation:

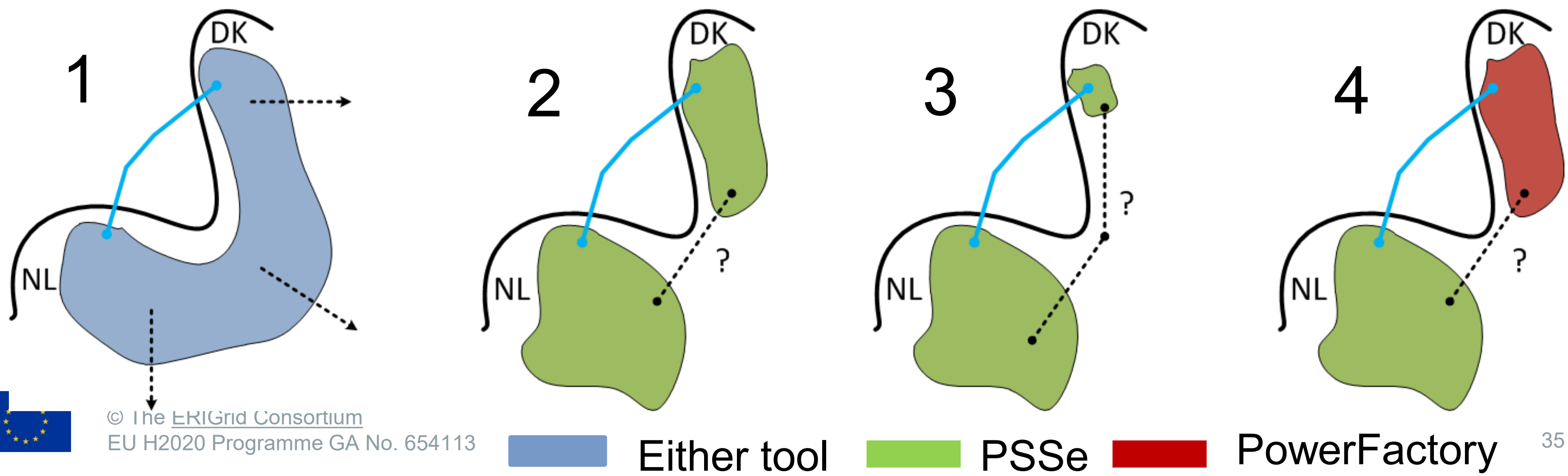
(105 s)

Active power:

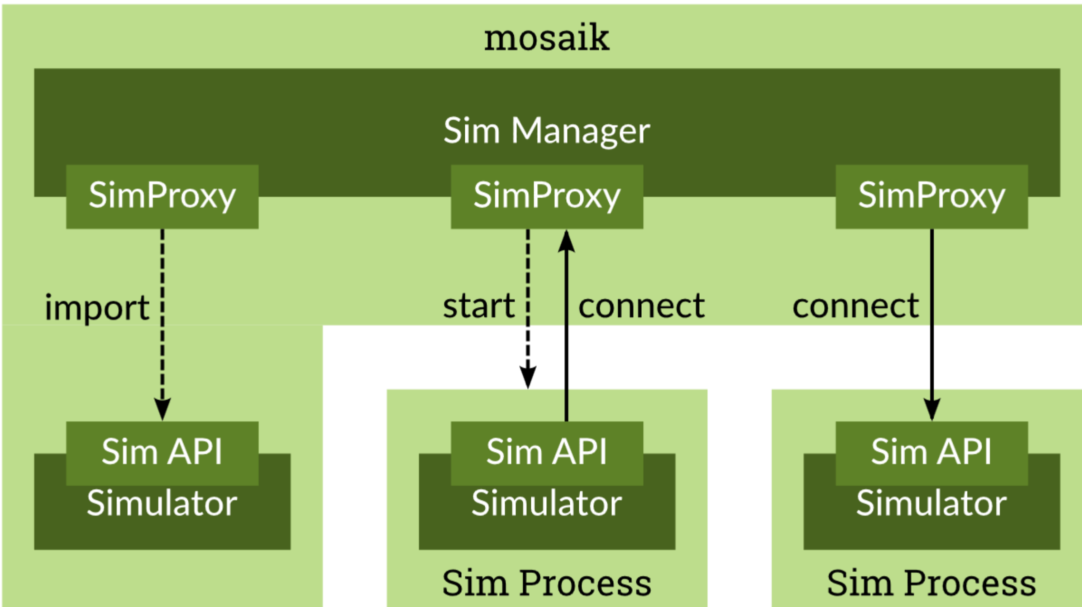


Co-simulation platforms: roll-out challenge, the COBRACable case

- Separate (dynamic) models of NL and DK
- NL: PSSe, DK: Powerfactory
- 4 options
  1. Integrate both models and perform monolithic simulation in PowerFactory or PSSe
  2. Migrate one grid model to the other simulation package
  3. Include simplified dynamic equivalent of the DK/NL grid model into the other
  4. Keep both models as such and perform a co-simulation between PowerFactory and PSSe
- Setting up co-simulation quite involved, knowledge at just a couple of persons, multiple licenses needed



Mosaik



https://mosaik.offis.de/

Characteristics:
- Python
- Socket-based
- JSON message encoding
- Discrete event scheduler

- From Mosaik to simulator:
  - init()
  - create()
  - step()
  - get_data()

- From simulator to Mosaik:
  - get_progress()
  - get_related_entities()
  - get_data()
  - set_data()

Pros: easy to use.

Cons: limited functionalities.

Summary of mosaik in ERIGrid

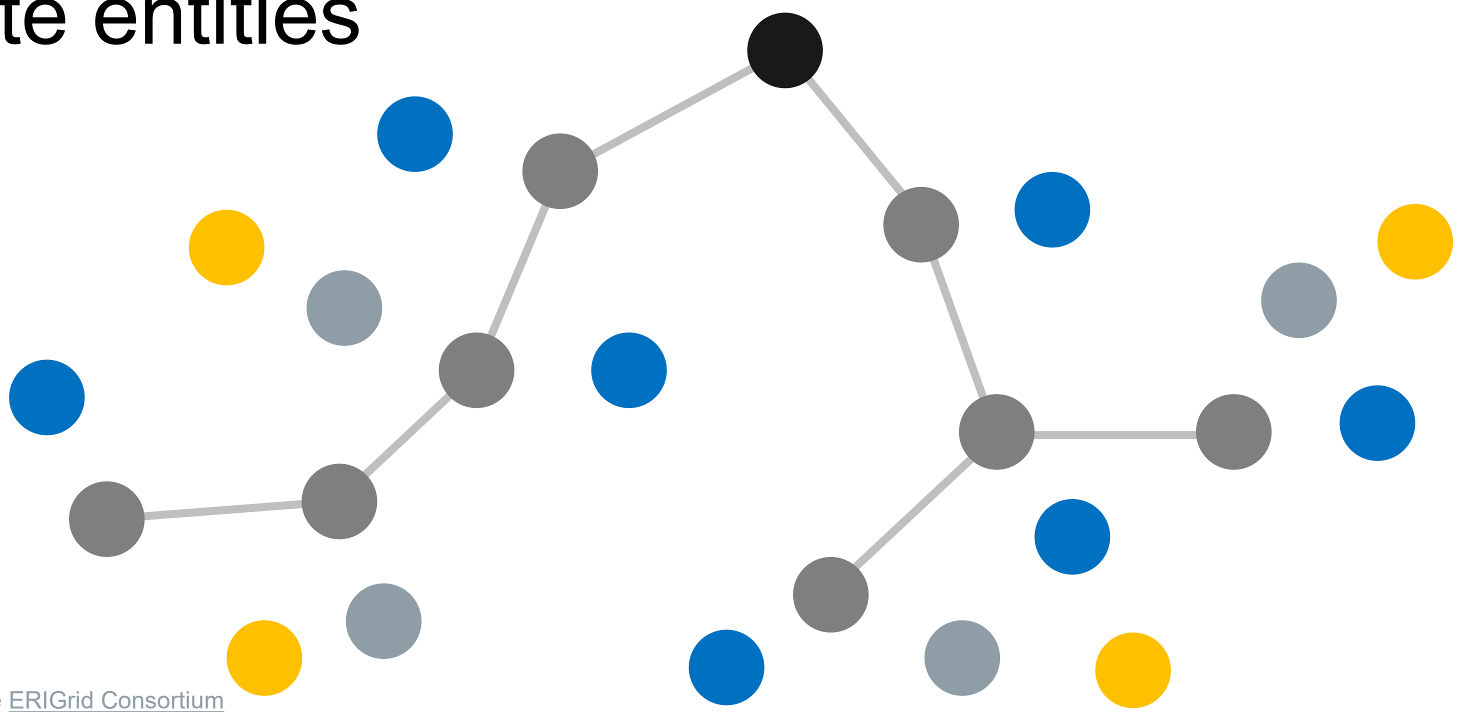
- Mosaik is a co-simulation tool
- Main co-simulation functionalities:
  - Organize data exchange
  - Synchronization
- Main use cases:
  - Create scenario
  - Connect simulators

Scenario Creation

List and initialize simulators

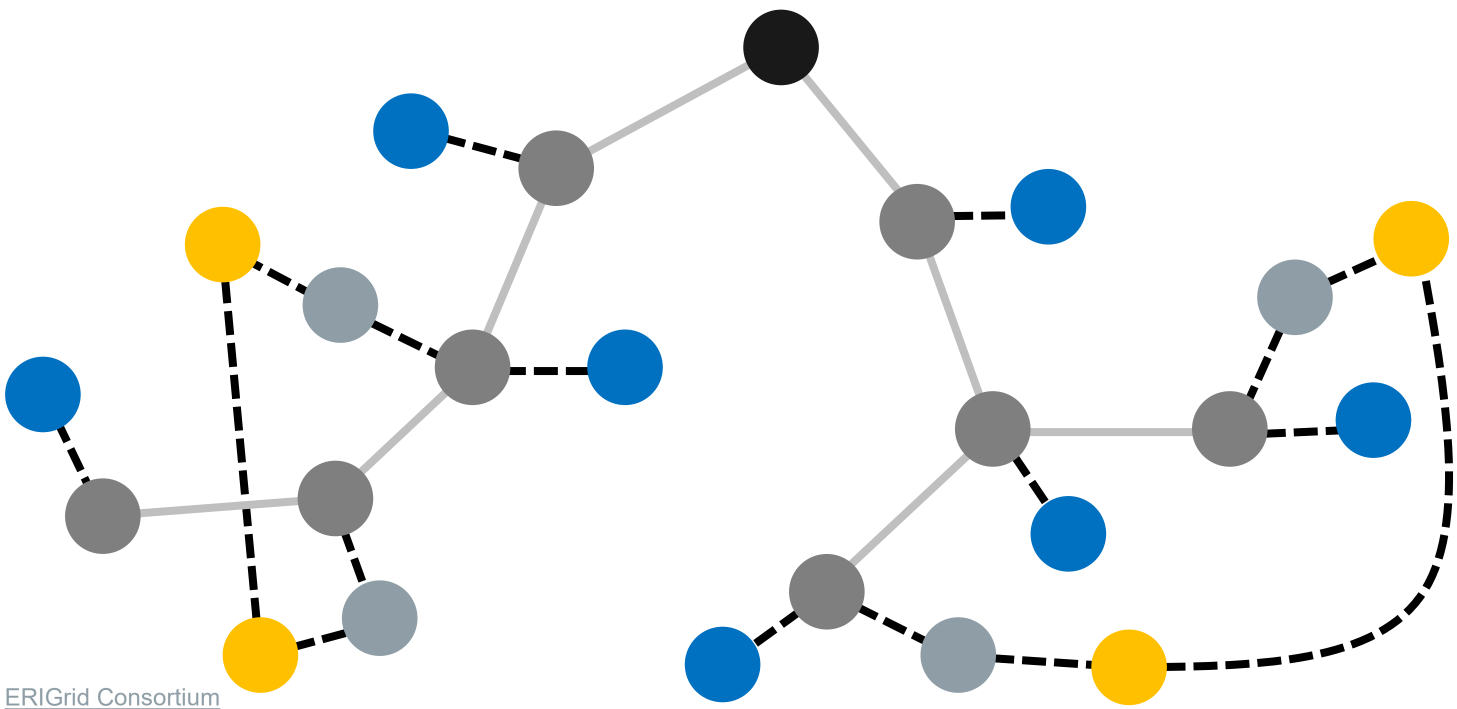
Scenario Creation

Create entities

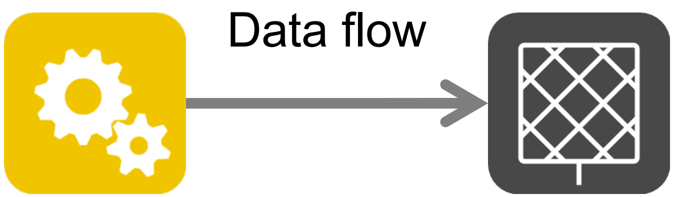


Scenario Creation

Connect entities



Cyclic Data Dependencies



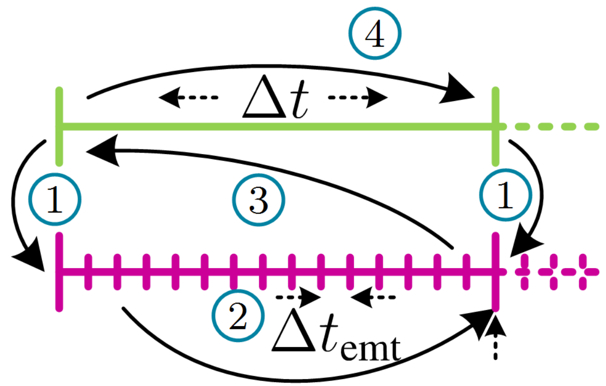
No problem

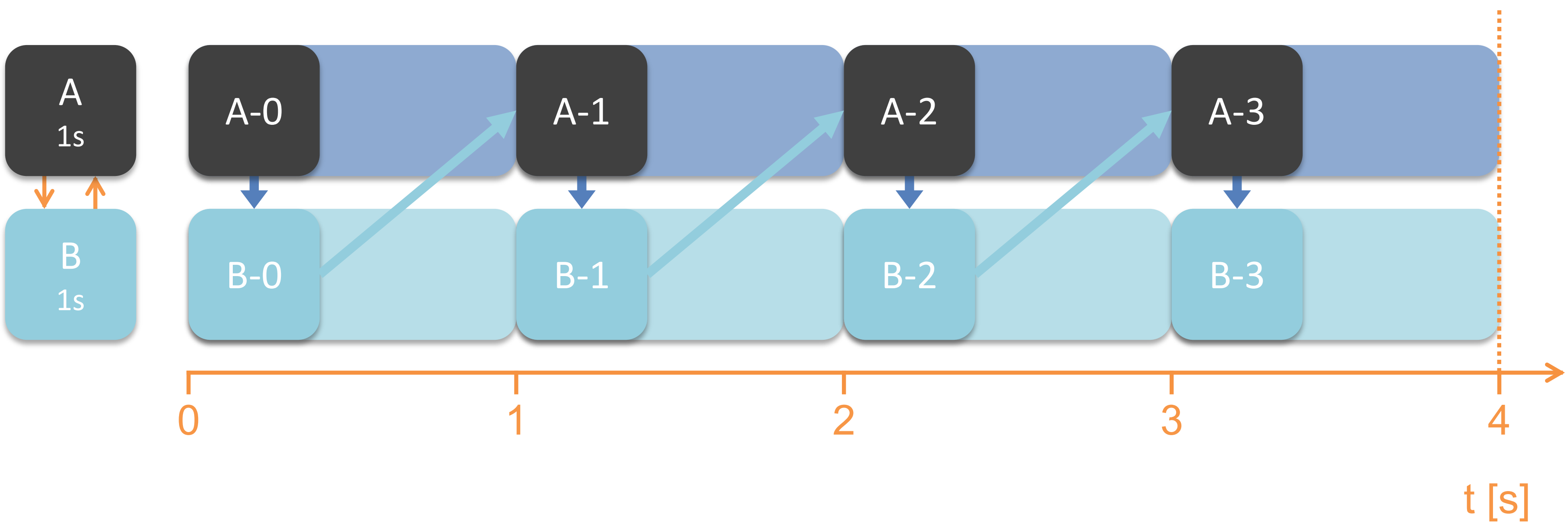
Problem
(Who comes first?)

Mosaik solution: *Asynchronous Requests*

Cyclic Data Dependencies

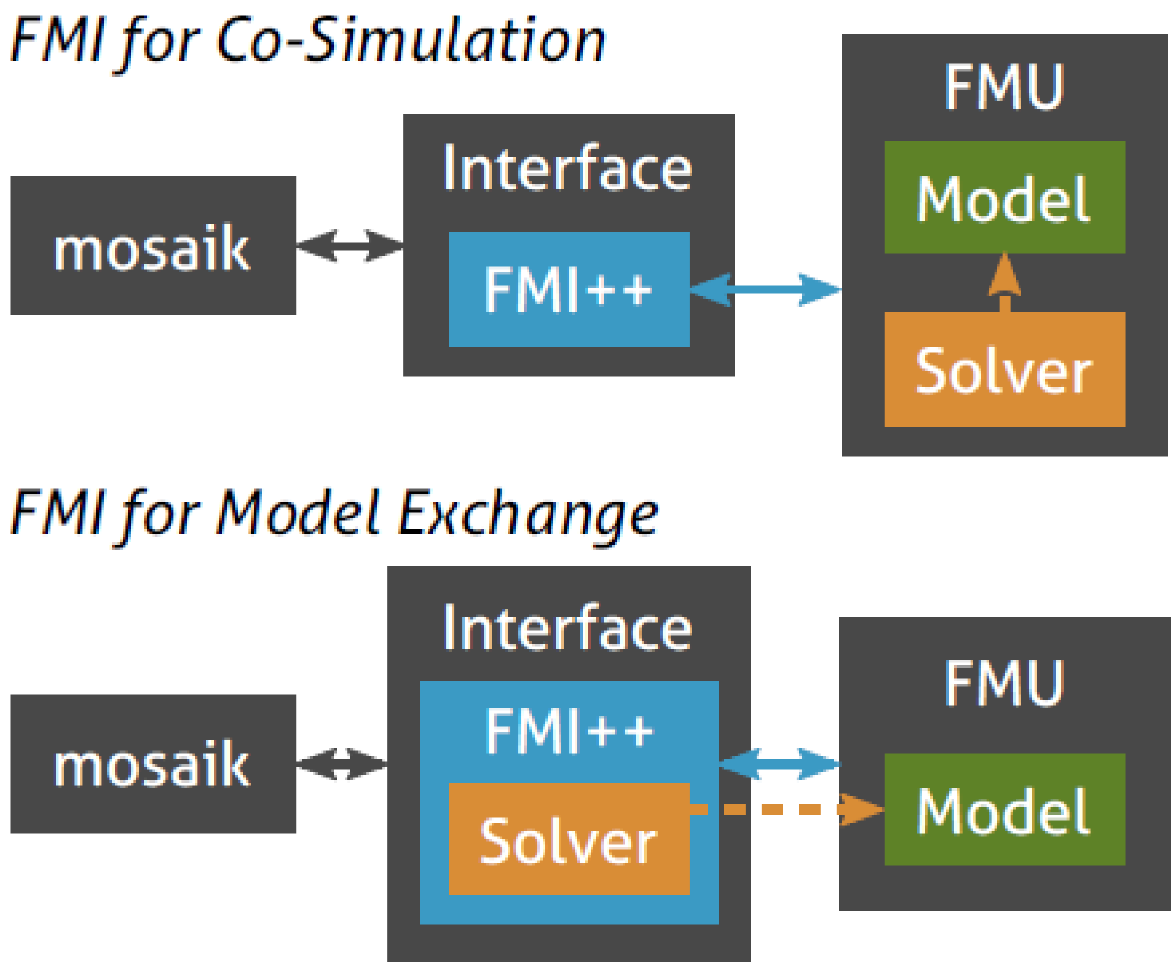
- Establish data connection in one direction
- Include asynchronous request
- Cycle is resolved via a shift in time





B.step(1) → 2

ERIGrid framework: FMI and mosaik



ERIGrid (JRA2) Co-simulation Models available as Open Source

- The work was done as a part of European Commission funded programme Horizon 2020 under the project European Research Infrastructure supporting Smart Grid Systems Technology Development, Validation and Roll Out (ERIGrid) [https://erigrid.eu].
- Interfaces and test systems developed available as Open Source models in github.

Links:

- Co-simulation assessment for continuous-time RMS studies (TC-1): https://github.com/ERIGrid/JRA2-TC1
- Combined Hardware and Software Simulation (TC-2): https://github.com/AIT-IES/FMITerminalBlock https://github.com/NabilAKROUD/OLTC_Arduino
- Signal-based Synchronization between Simulators (TC-3): https://github.com/ERIGrid/JRA2-TC3

Towards (co-)simulation as a service

ERIGrid 2.0:

- Improved Multi-domain co-simulation (bit of technology, further integration of standards)
- Improved and extended services for simulation based assessment
  - Middleware for distributed lab testing (user management, time resolution, remote graphical consoles, transport services)
  - Data as a service (proxy models for real-time and non-real time data sources)
  - Extended services for simulation environments (distributed test configuration, automated setup tests)
  - Simulation setup automation
  - Offline simulation as a service prototype (integrate existing tools into web-based environment)

Conclusions

- Energy system complexity can be mastered with co-simulations
- Important elements: proxy models, interfaces, master
- ERIGrid focused on:
  - FMI implementation for PowerFactory, PSSE, NS-3,
  - Compatibility mosaik and continuous time simulations
  - Testing scalability of co-simulation based assessment methods
- Next steps:
  - further standardisation of co-simulations
  - system assessment aaS
  - Apply co-simulation as a tool for lab-integration

EU H2020 Programme GA No. 654113

Thank you

Dr. Arjen A. van der Meer

ERIGrid 1.0 WP leader of JRA2: “Co-simulation based assessment methods”

ERIGrid 2.0 WP leader of JRA3: “Improved and Extended Services (RI integration, coupling, and automation)”

Programme Manager PowerWeb Institute, part of Delft University of Technology

a.a.vandermeer@tudelft.nl