Optimization of FMUs Assemblies

SEPTEMBER 2025 – Nicolas LAURENT

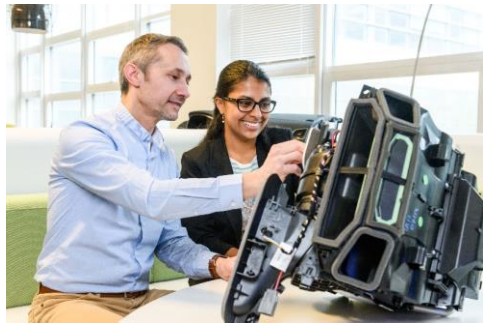
Renault group key figures

Group revenues
(€ billion)
2024 **56.2 B**

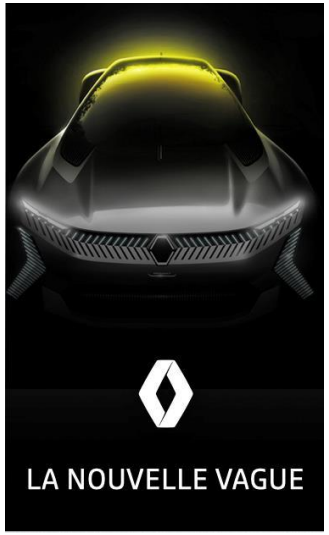
98 000
employees
in 36 countries

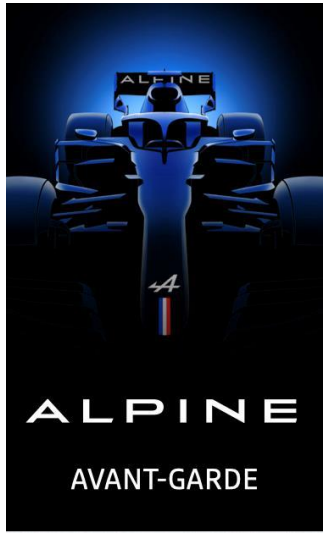
2.2 M
vehicles
sold
in the world
in 2021

Renault Group



4 brands





Context and automotive challenges

Energetic transition
(Electrified vehicles)

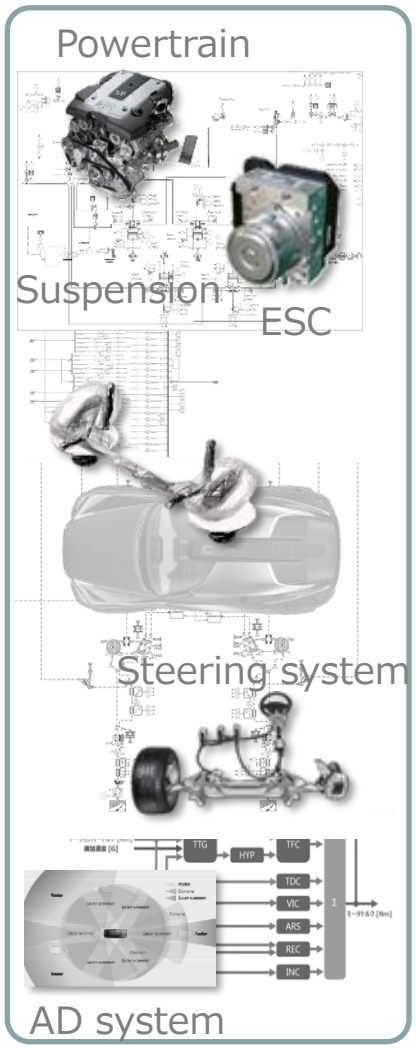
Software
(Connected & Autonomous vehicles)

Regulatory / compliance

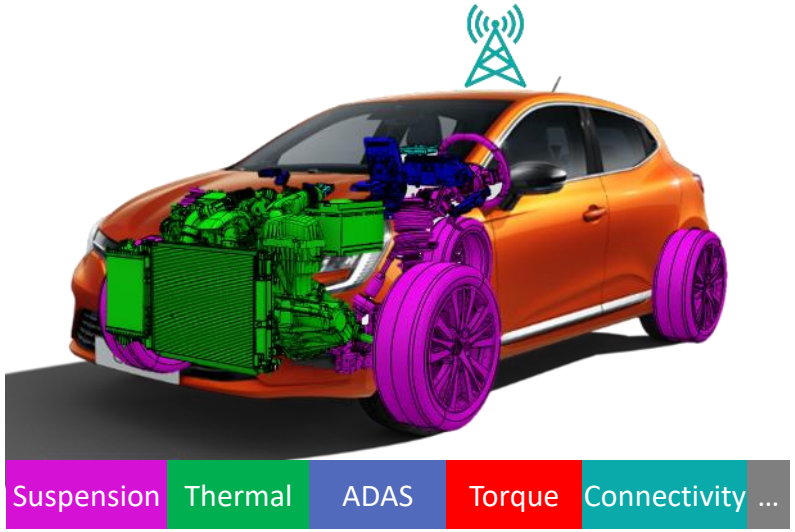
Vehicle are more and more complex, time to market is shorter and shorter :

SIMULATION TO SUPPORT DEVELOPMENT AND VALIDATION OF VEHICLE

Collective Engineering Models and software

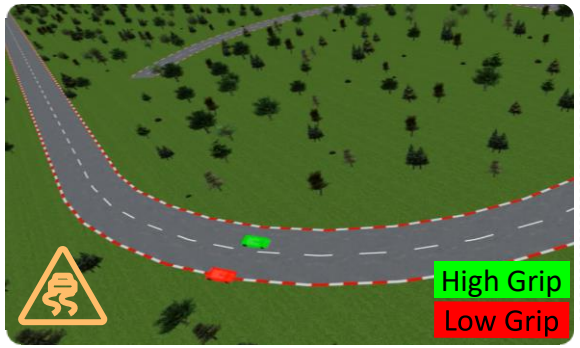


Connect the 0D-1D models together in order to build a vehicle digital-twin…

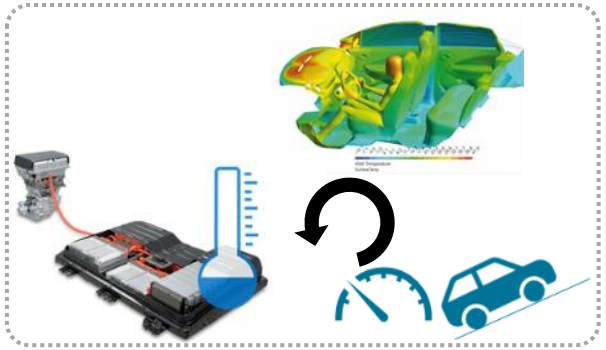


… to simulate more and more numerous and complex test cases

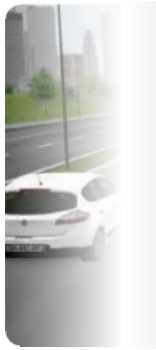
Autonomous Emergency Braking



Torque management



Thermal management

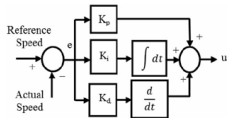


…

Simulation Platform

Involved artefacts

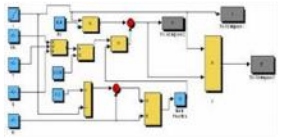
Human Driver



Driver Model

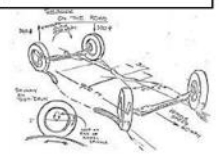
=

CONTROL



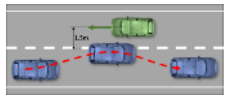
+

PHYSICS



Driver & customer

Terrains



Scenarios

Digital Vehicle

Simulation environment

Simulation Platform architecture

Requirements

Results

Models & Embedded software

Scripts & Simulation tools

Assessment

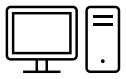
Means

Simulators

XR simulation

Quantify of interest

KPI



Work station

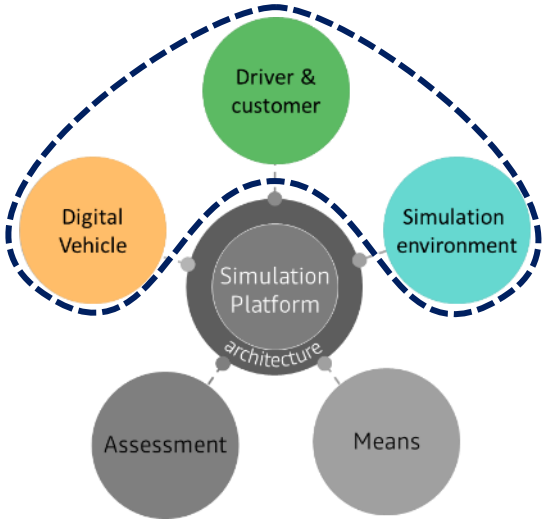
Cloud

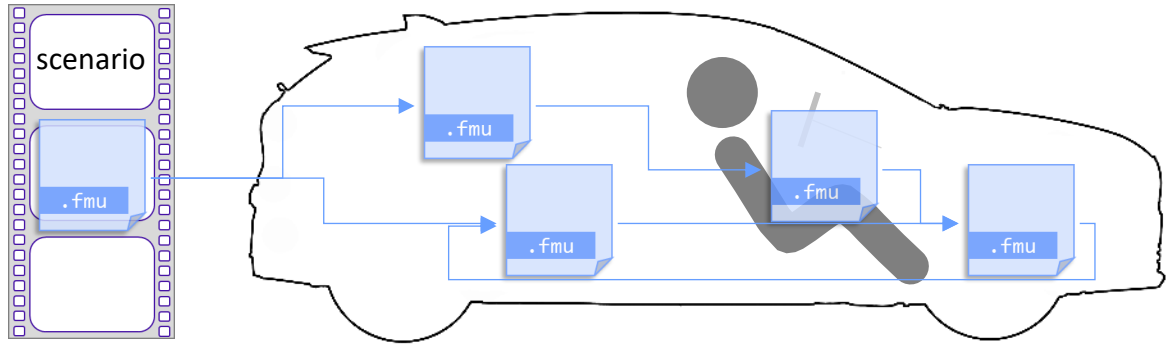
Assembly of simulation Platform: FMI in Co-simulation mode

OUR USAGE OF fmi STANDARD

- **Co-Simulation** FMU: simple to produce, simple to use, IP protection
- Streamline communication with suppliers : a **neutral** and **widely supported** format
- May **optimize license** costs
- More and more involved actors (software editors, Industrial, Tiers 1, …) to enforce **inter-operability**
- **Robustify interfaces** between Simulation means and simulations assets (Digital Vehicle, Drivers and Simulation Environment)

Note: **FMI version 2** is currently more supported than version 3 among tools and our suppliers

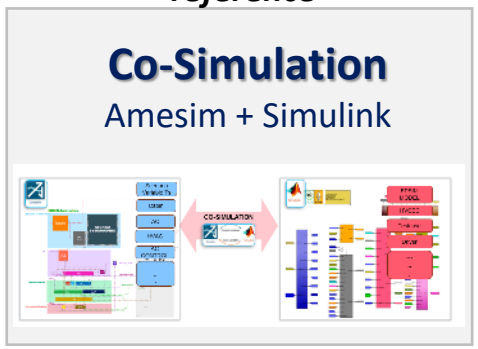




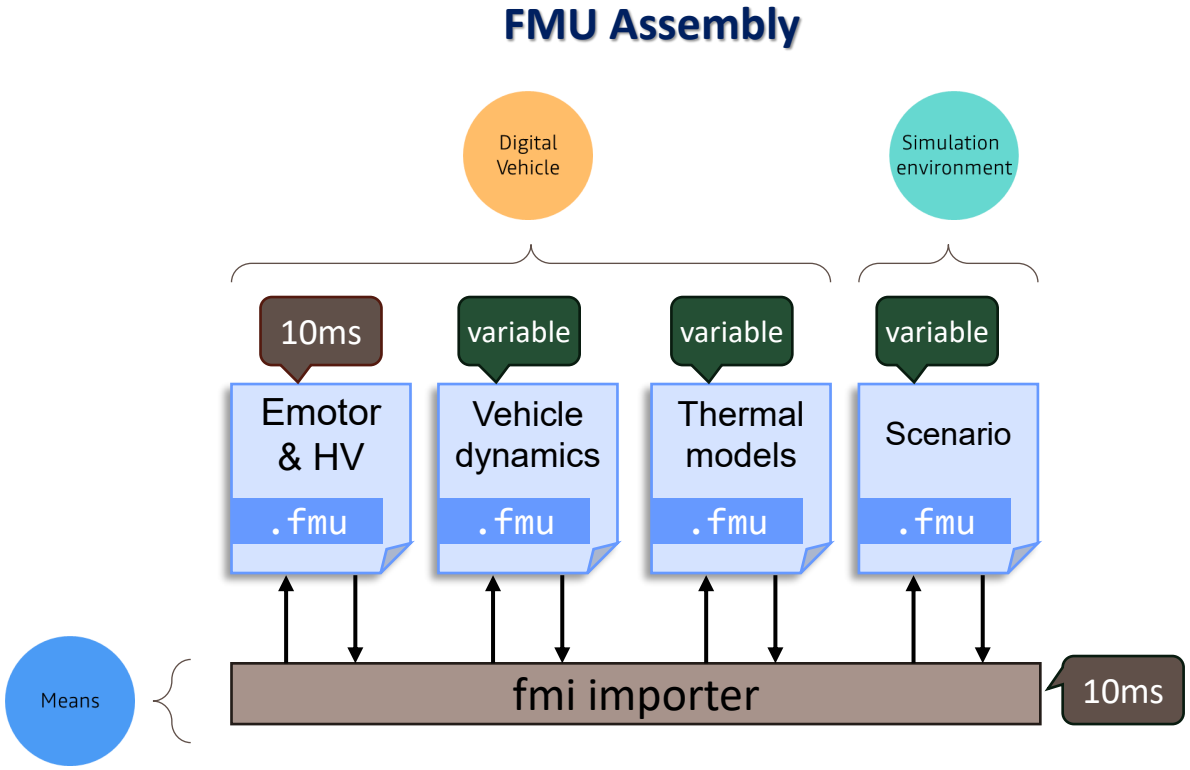
FMU Assembly

Thermal Management example

reference



vs.



Former simulation process

Ad-Hoc assembly involving 2 modeling tools at runtime
variable time step solvers

Compute time = 4.2h

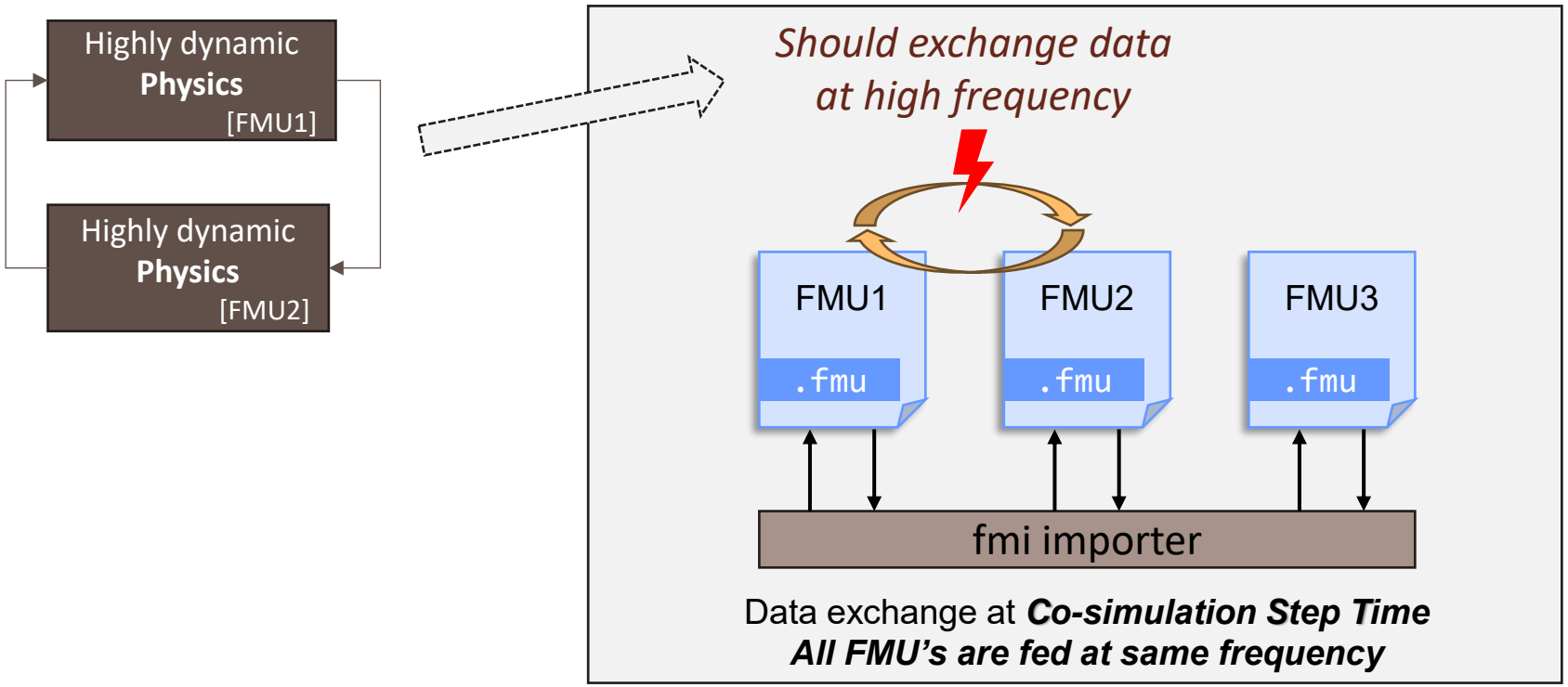
Unified process

Low dependency on modeling tools
Fixed co-simulation time step

Compute time = 48h

Co-simulation step size greatly affects simulation performance and accuracy.

Tightly linked models within FMUs assembly can lead to performance issue



Setting lower ***Co-simulation Step Time*** will solve FMU feeding frequency issue but leads to overall poor performances.

↘ ***Co-simulation Step Time*** ➤ ↗ Numerical ***accuracy*** & ↘ CPU ***Performances***

↗ ***Co-simulation Step Time*** ➤ ↘ Numerical ***accuracy*** & ↗ CPU ***Performances***

Tightly linked models within FMUs assembly: ways to improve performances

VARIABLE CO-SIMULATION STEP TIME

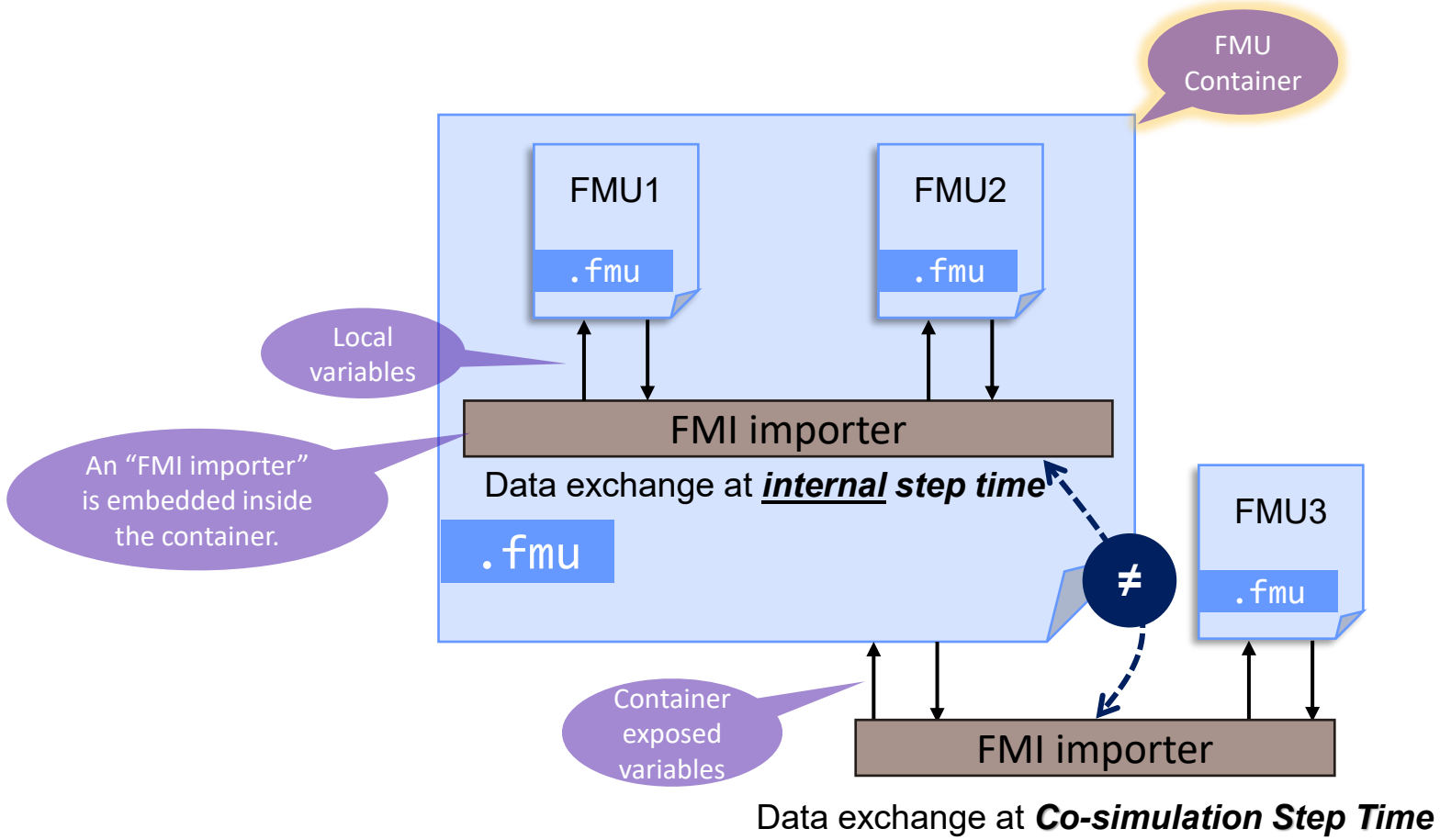
+ **Most efficient** tradeoff between performance and accuracy
- Algorithm to dynamically set *Co-Simulation Time Step* are not trivial
- Some FMU may not support variable *Co-Simulation Time Step*
- *Not applicable on RealTime simulation means*

MULTIPLE CO-SIMULATION STEP TIME

+ May be a **good tradeoff** between performance and accuracy
- Require additional effort for architecture design and to build the assembly
+ May enable multi-threading to improve performance

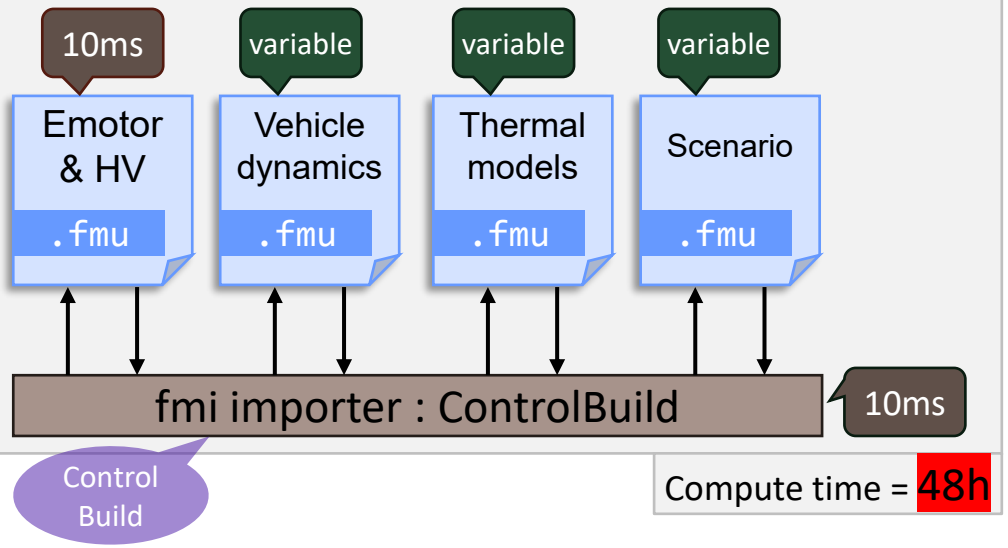
Can be done through FMU Containers

Multiple Co-Simulation Step Time : FMU Container



Thermal Management application

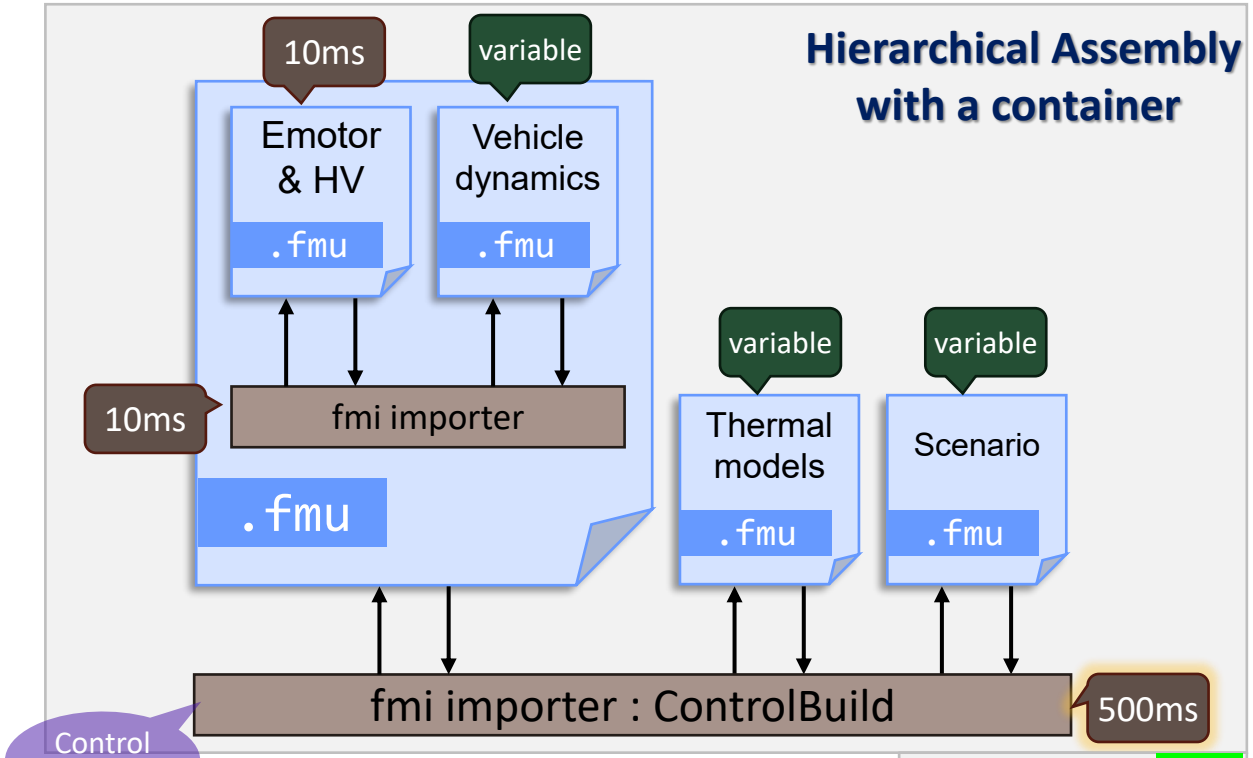
Flat FMU Assembly



Compute time = 48h

vs.

Hierarchical Assembly with a container



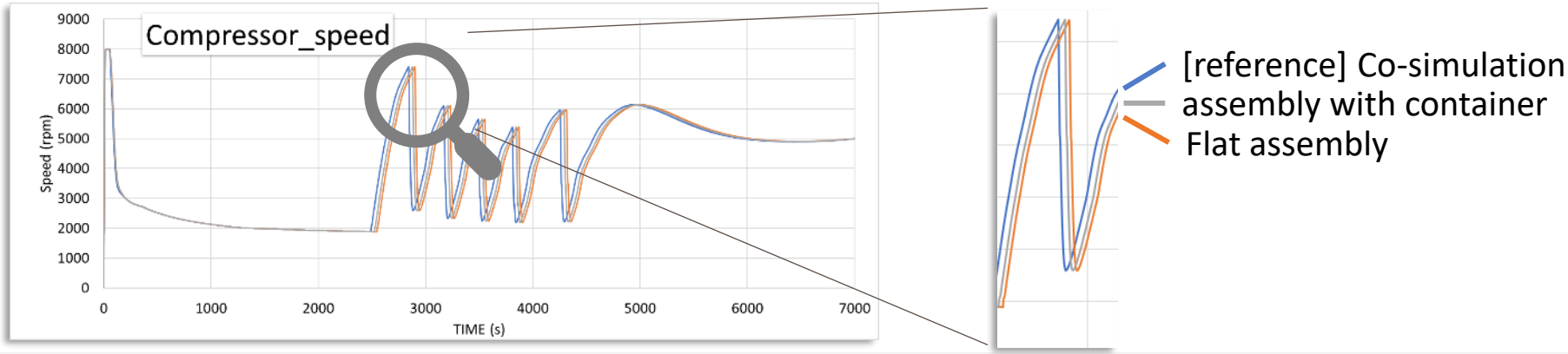
Compute time = 4.3h

reference

Co-Simulation

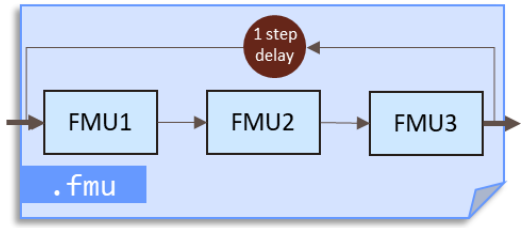
Amesim + Simulink

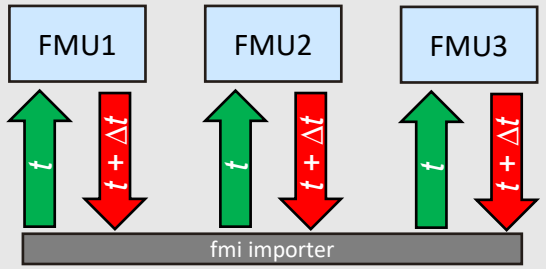
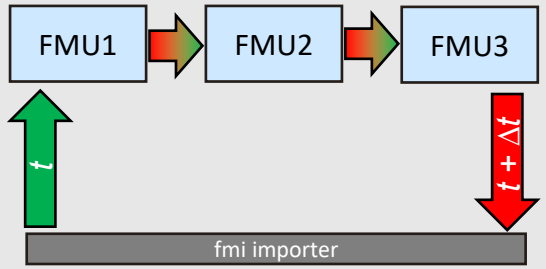
Compute time = 4.2h



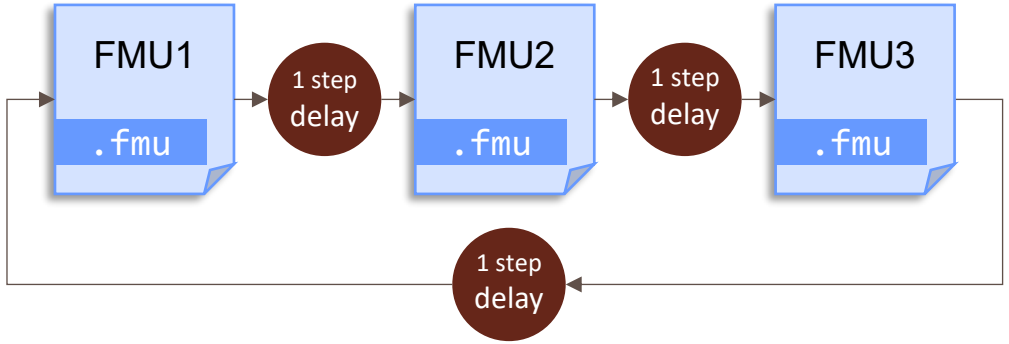
Execution of simulation

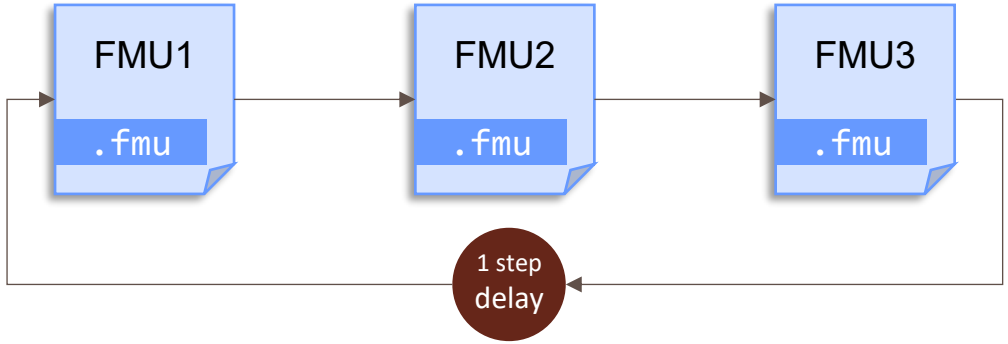
2 STRATEGIES FOR EXECUTION



Parallel	Sequential
	
+ Scalable : enable parallelization - Imply communication delays	- Order should be defined + minimize communication delays

IMPLIED DELAYS BETWEEN FMU

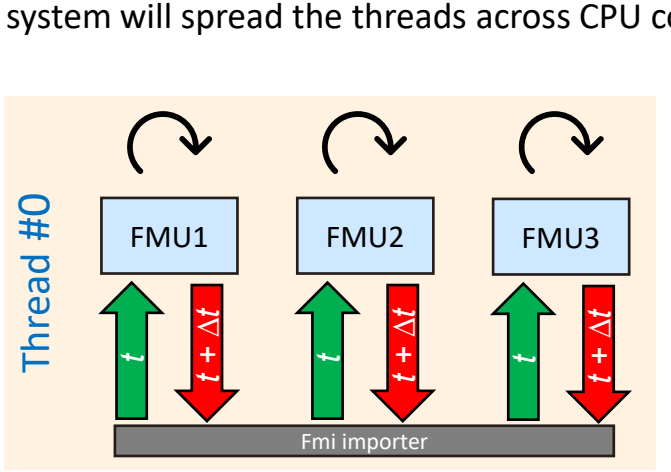


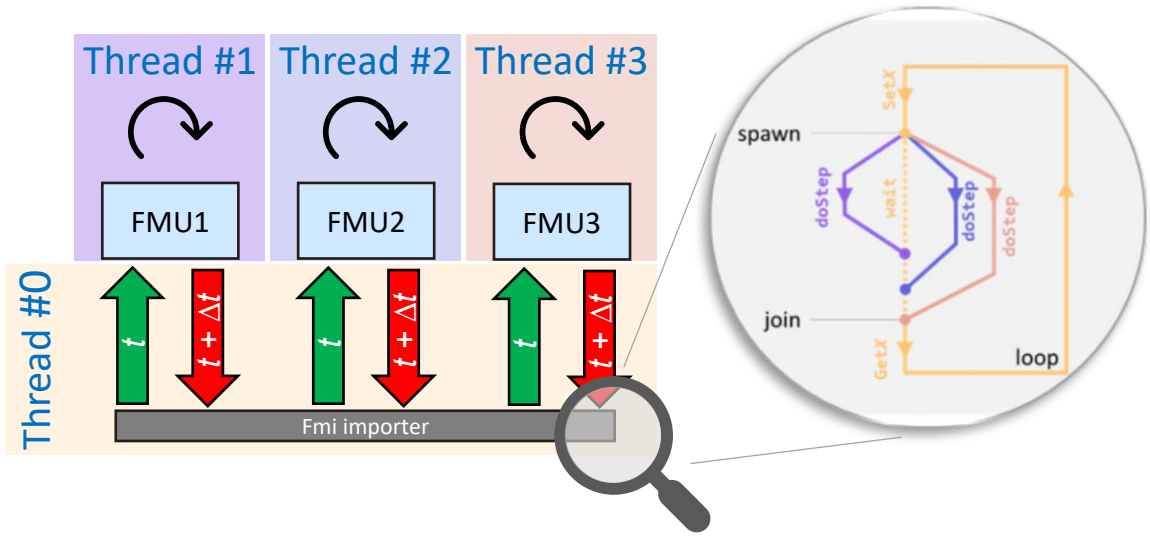


Parallel mode: Multi-thread *vs* Mono-Thread

STRATEGY

Operating system will spread the threads across CPU cores

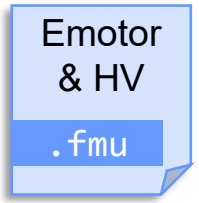




USE CASE

Potential benefits grow when simulate in parallel FMU's which eat approximately the same amount of CPU cycles.

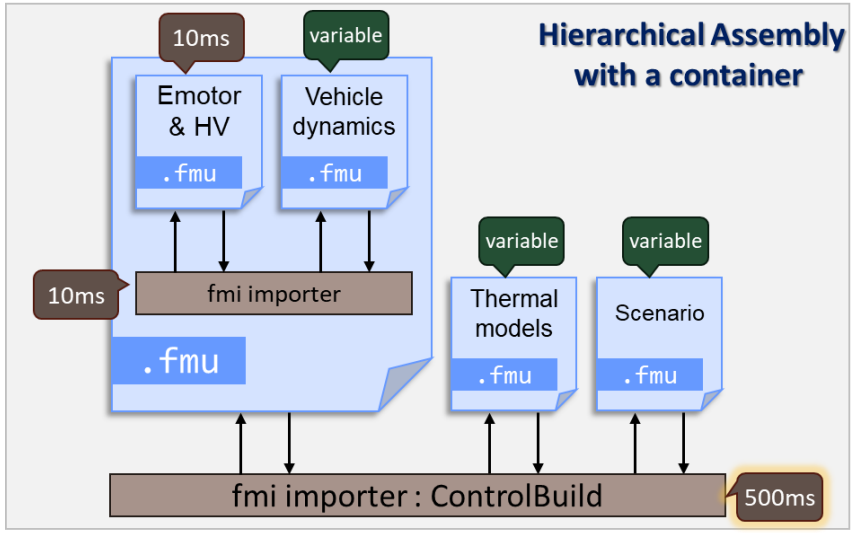
In this example, speed up will be limited

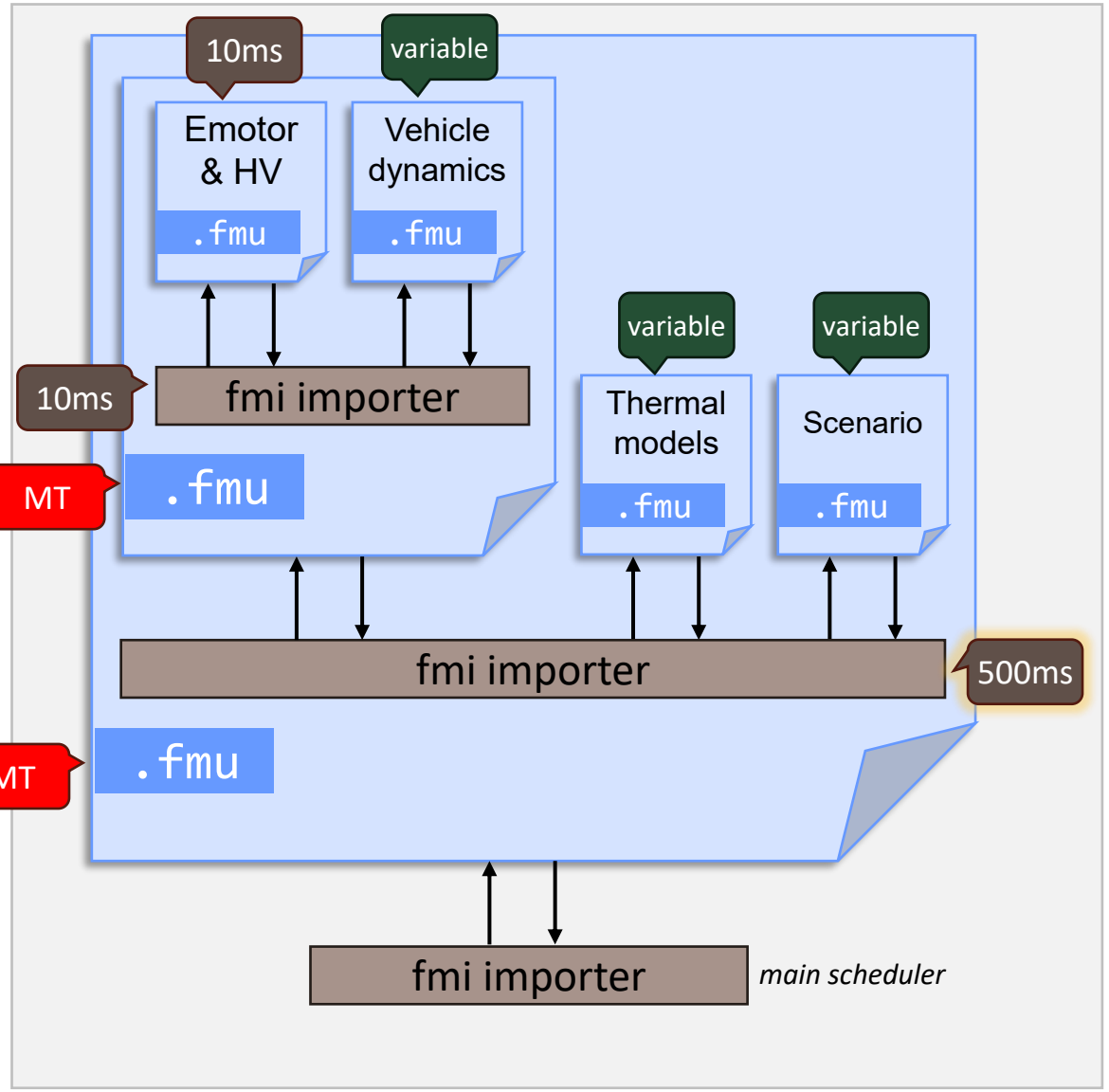



```
RT Ratio: 1.68 (simulation is FASTER than reality)
```


```
RT Ratio: 106.38 (simulation is FASTER than reality)
```

Mutli-thread and multi-level container

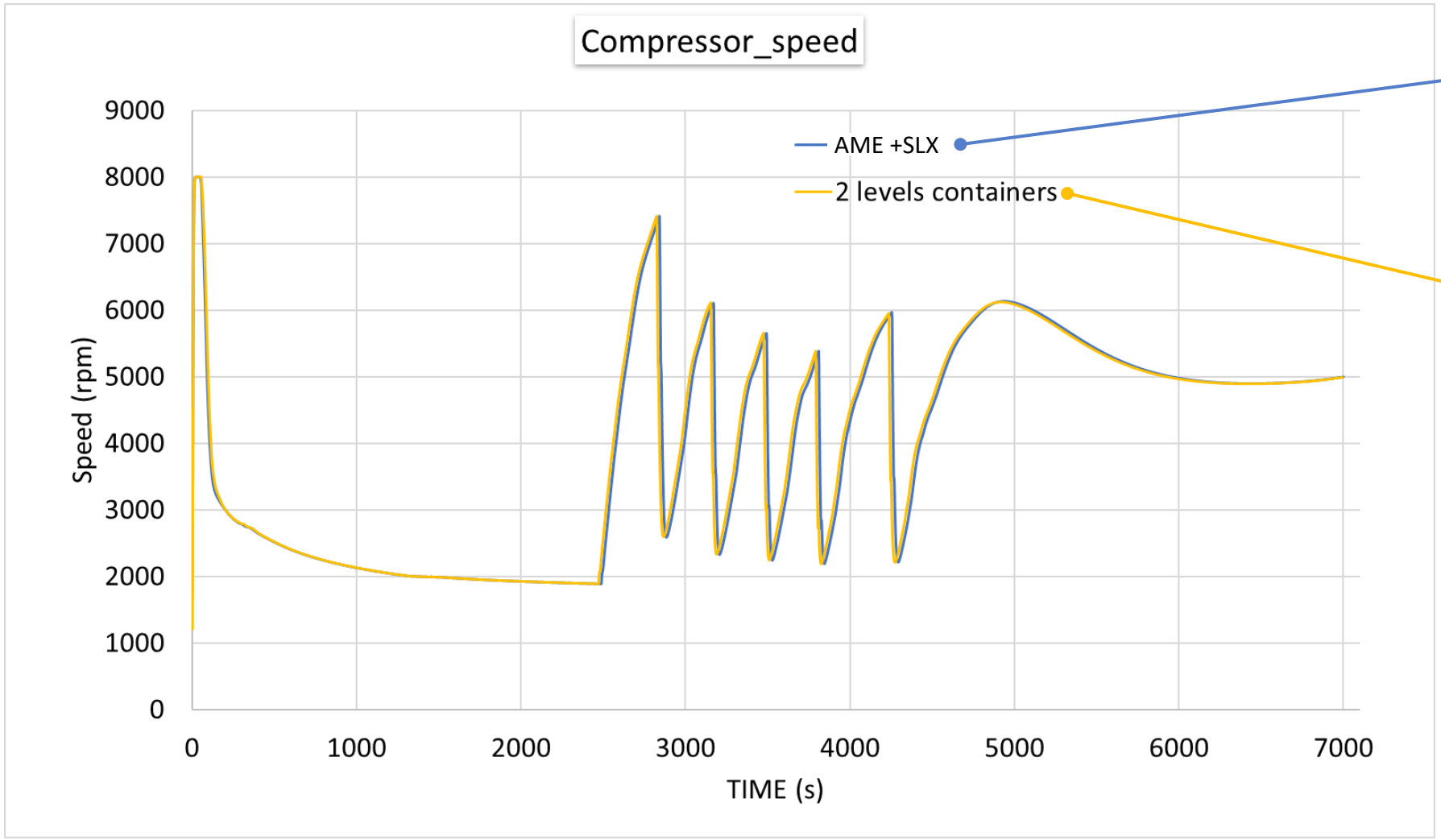




Hierarchical Assembly with 2 containers

Enable muti-thread for all FMUs with a second level of containers

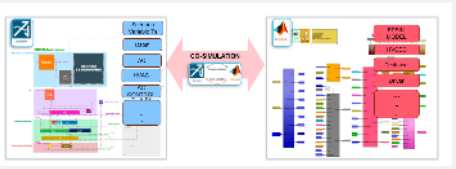
Mutli-thread and multi-level container : results compared to reference



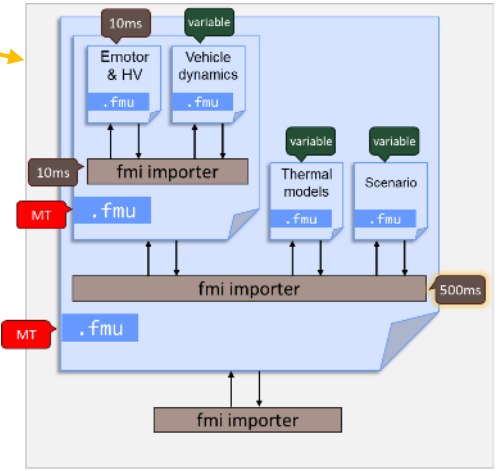
reference

Co-Simulation

Amesim + Simulink



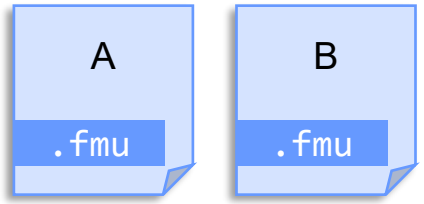
Compute time = 4.2h



Hierarchical Assembly with 2 containers

Compute time = 3.7h

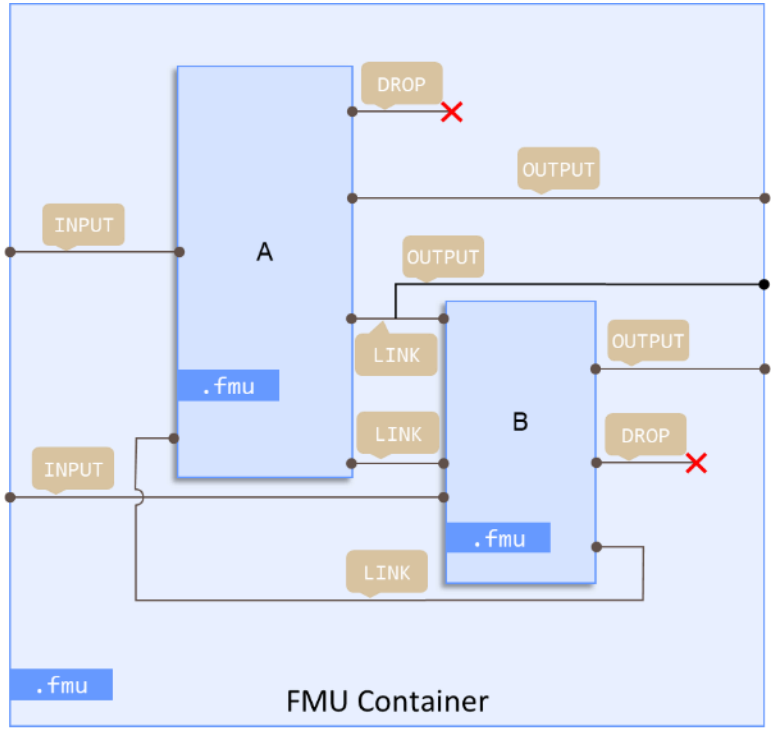
Containers made easy with FMU Manipulation Toolbox



Rule	From		To	
	FMU	Port	FMU	Port
FMU	A.fmu			
LINK	A.fmu	portOutA	B.fmu	portInB
LINK	A.fmu	portAandB	B.fmu	portAandB
INPUT		portInput	A.fmu	portInA
INPUT		portInA	A.fmu	portInA
OUTPUT	A.fmu	portAOut		portOutput
OUTPUT	A.fmu	portAOut		PortAOut
DROP	A.fmu	portOutA		
START	A.fmu	portInA	3.1415	

Routing file

fmucontainer command line



json or SSP files can be used to define a multi-level container and build it in one step.

pypi.org/project/fmu-manipulation-toolbox

github.com/grouperenault/fmu_manipulation_toolbox

BSD-2-Clause license

Future plans:

- Link with MBSI (Catia Magic)
- FMI-3.0 portage
- Layered standards support

FREE DOWNLOAD


Thank you