CREATORS

SUSTAINABLE PLACES 2021

Sep. 28 - Oct. 1, 2021 | Rome, Italy

Advances in distributed generation sources digital twins design

Ivan Todorović
Ivana Isakov
Boris Sučić
Dušan Čohadžić
Aleksandar Kavgić

Digital twinning

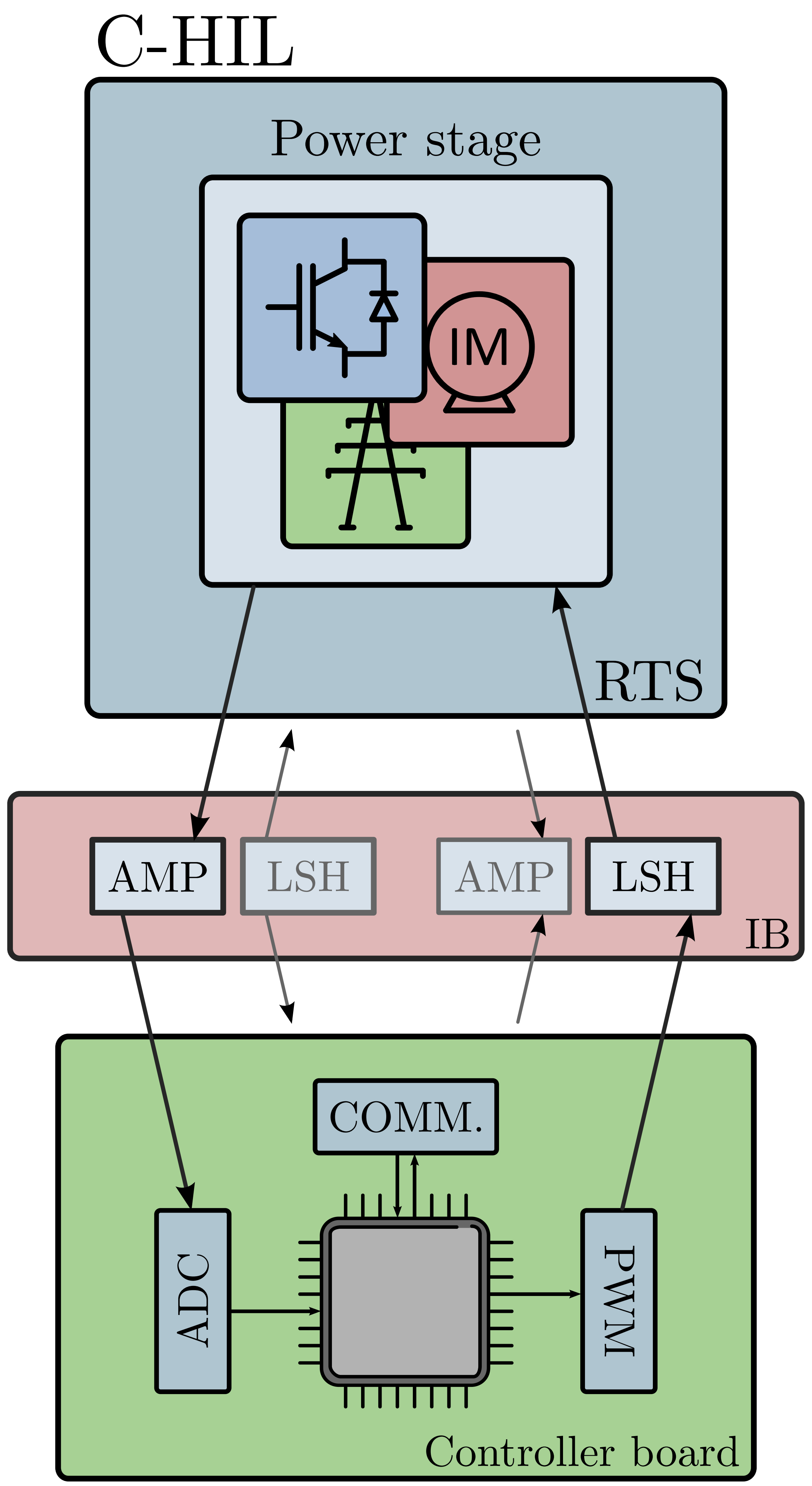
Why do we need it?

- Minimize hazards of LEC/CES usage
- Maximize profit and minimize financial losses
- Facilitate the development of new features
- Explore new business models and revenue streams
- Enable grassroots incentives

To put it simply:

We want to know what is going to happen with CES!
(as much as possible)

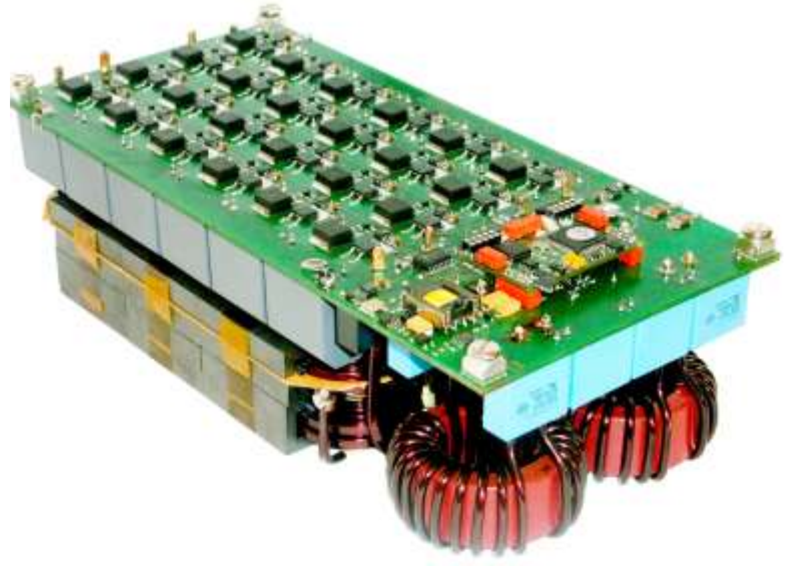
We do not want to assume and just hope for the best!

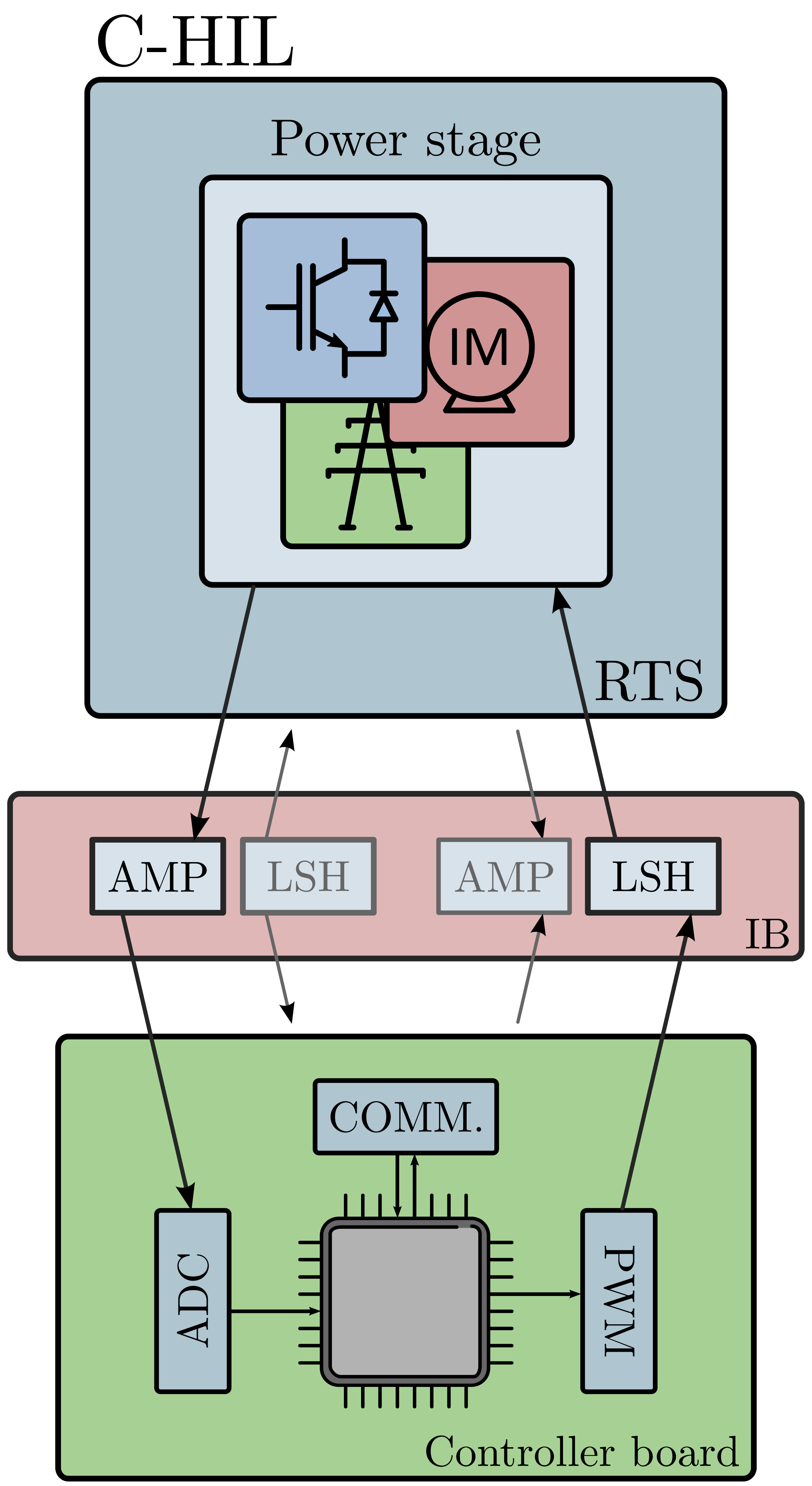


Digital twinning

What is it?

- Digital replica of a multi-domain systems (electrical, mechanical, chemical, thermal, etc.)
- To be executed slower, faster or in real-time

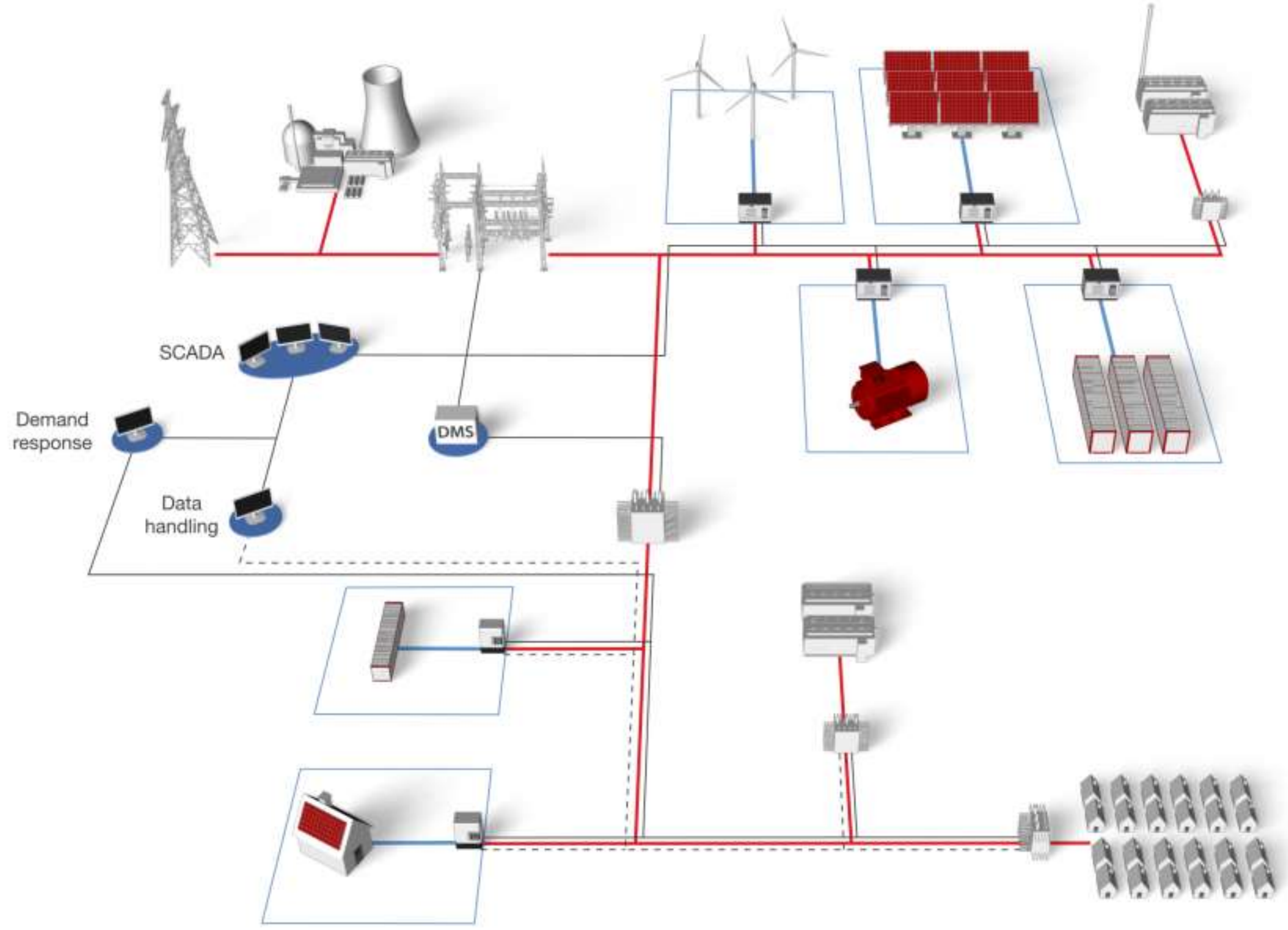


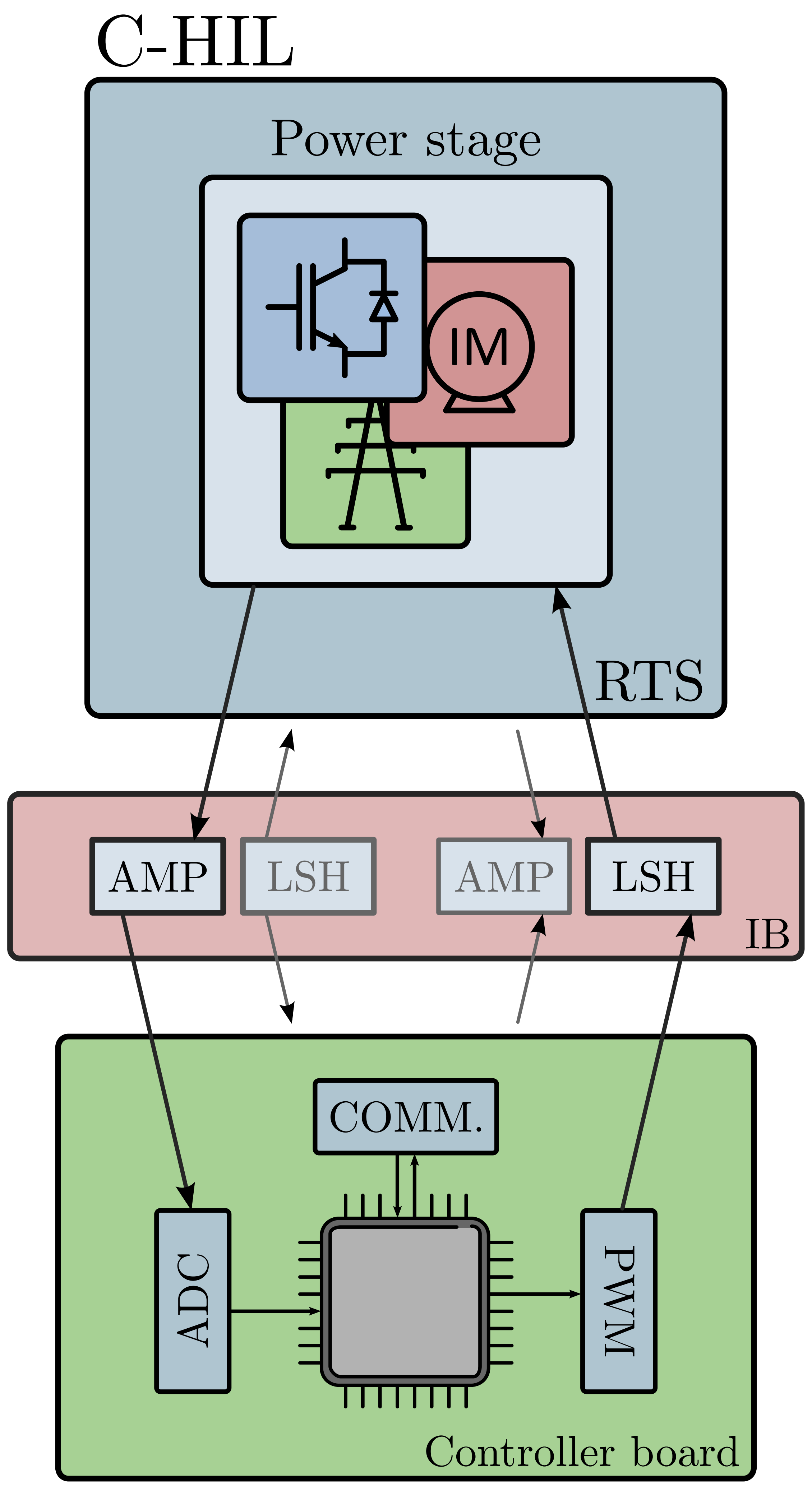


Digital twinning

What is it?

- Digital replica of a multi-domain systems (electrical, mechanical, chemical, thermal, etc.)
- To be executed slower, faster or in real-time

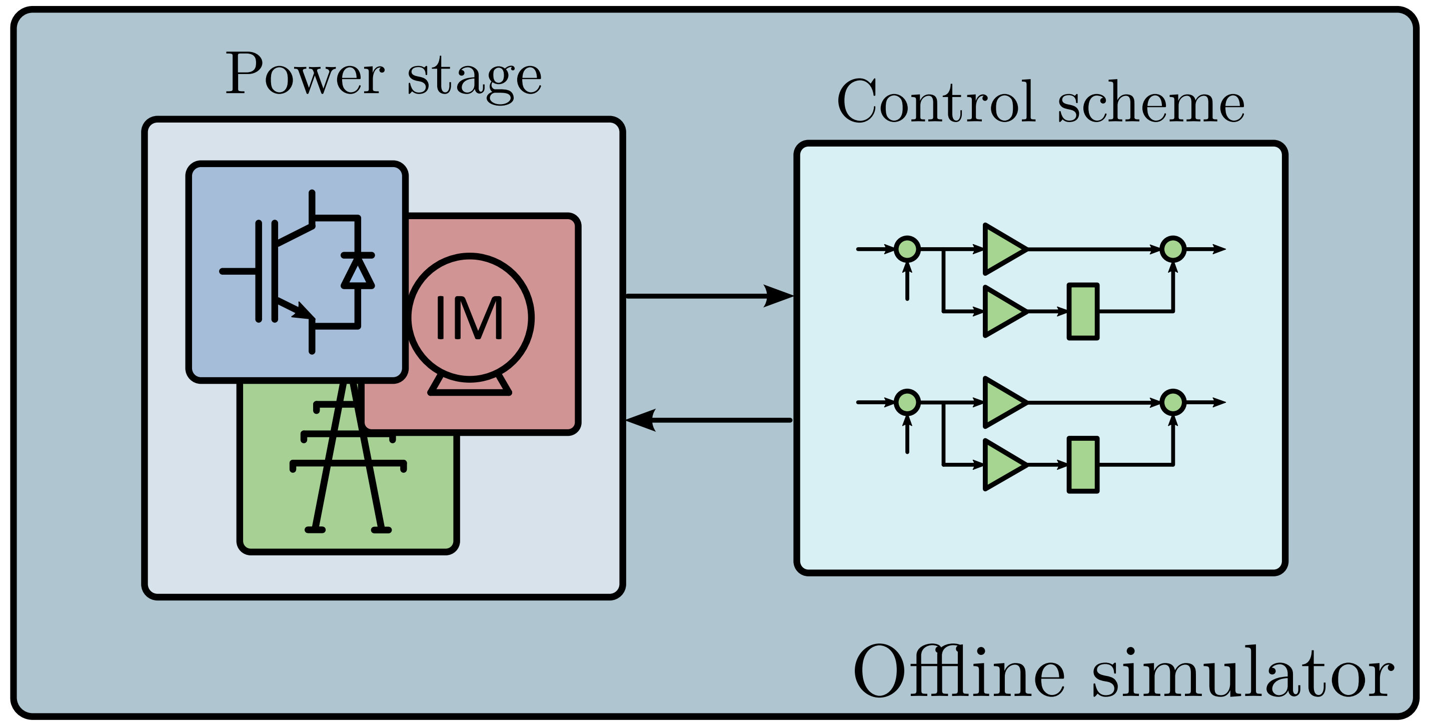




Traditional

Digital twinning

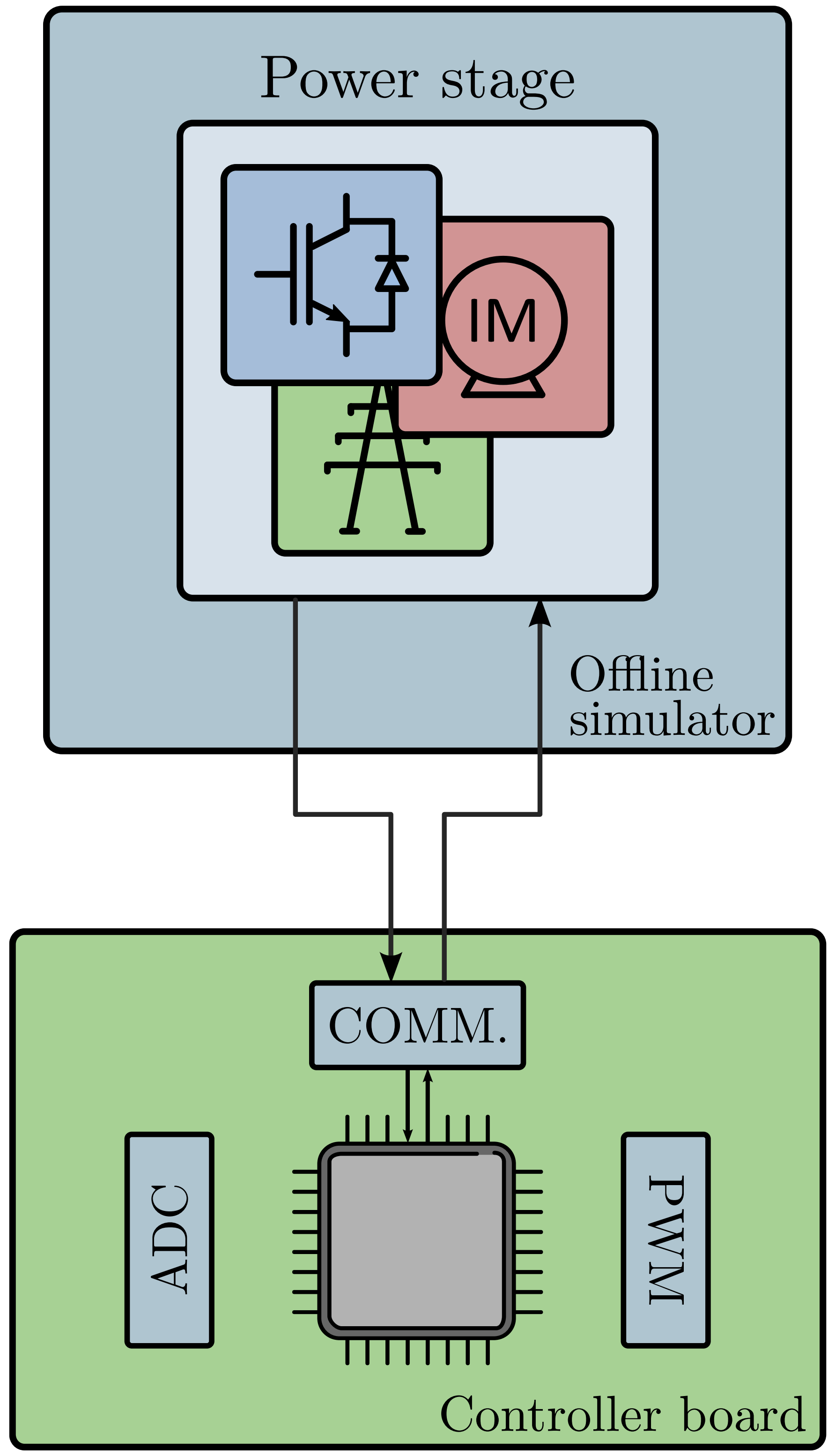
MIL



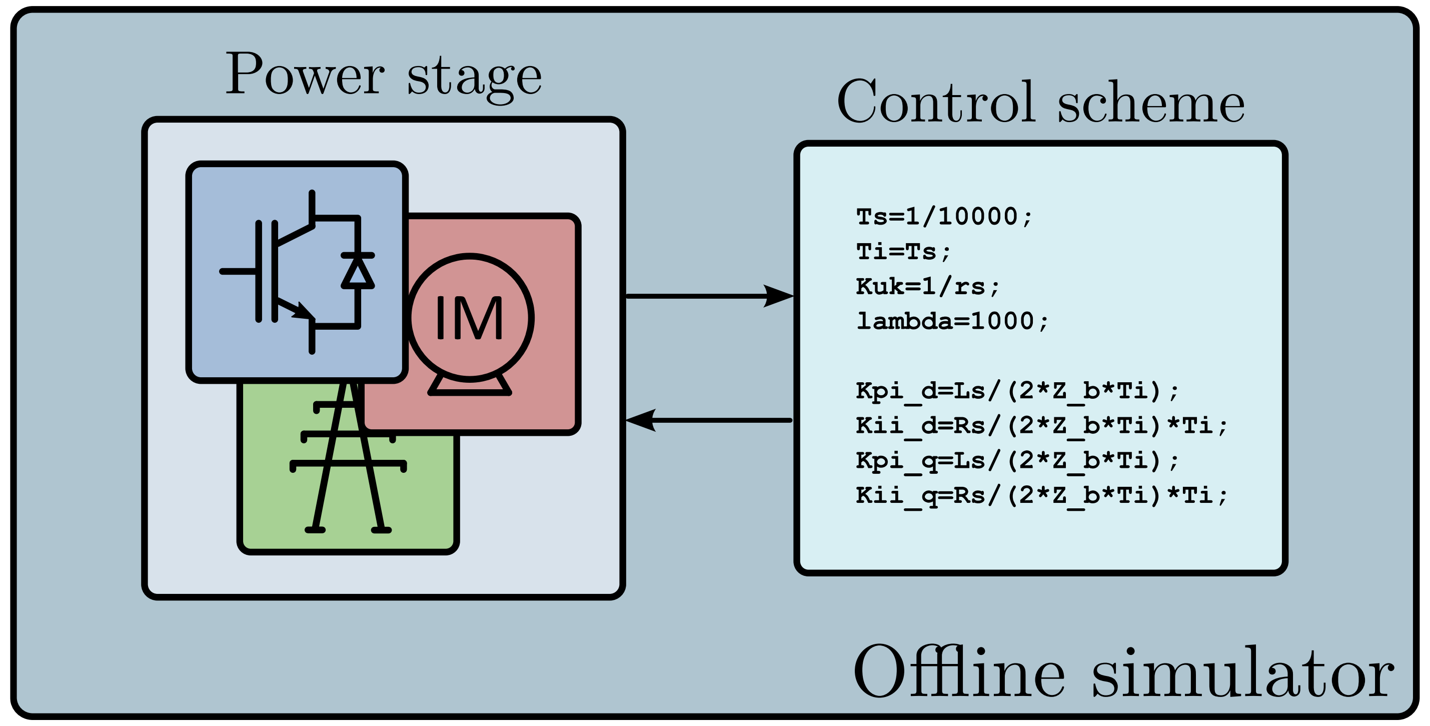
Model in the loop

- PC-based simulators
- Detailed modeling
- Safe & Slow
- Developed community and libraries
- Crude idea of how the end product will behave

PIL



SIL



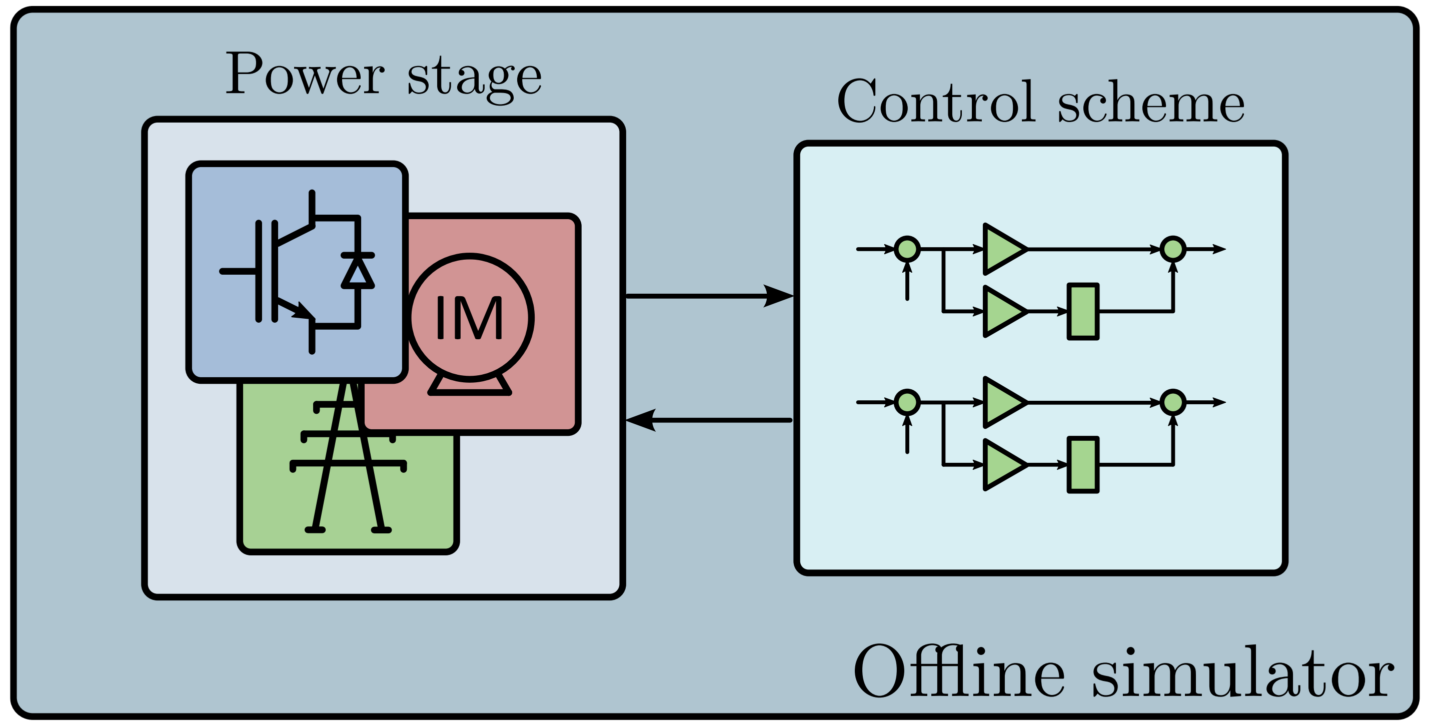
Software in the loop

- PC-based simulators
- Safe & Slow
- Dedicated compiler necessary
- Convenient way of checking the code organization and syntax
- Still, crude idea of how the end product will behave

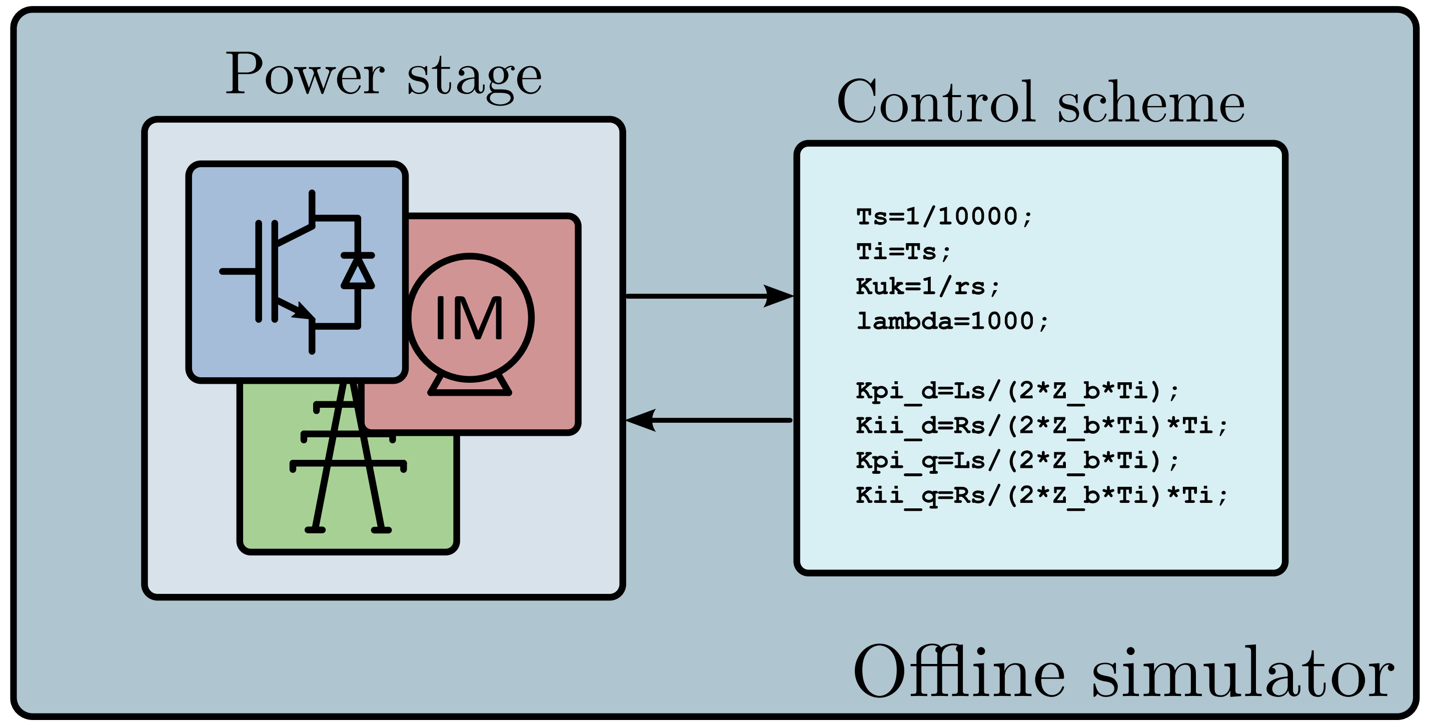
Traditional

Digital twinning

MIL



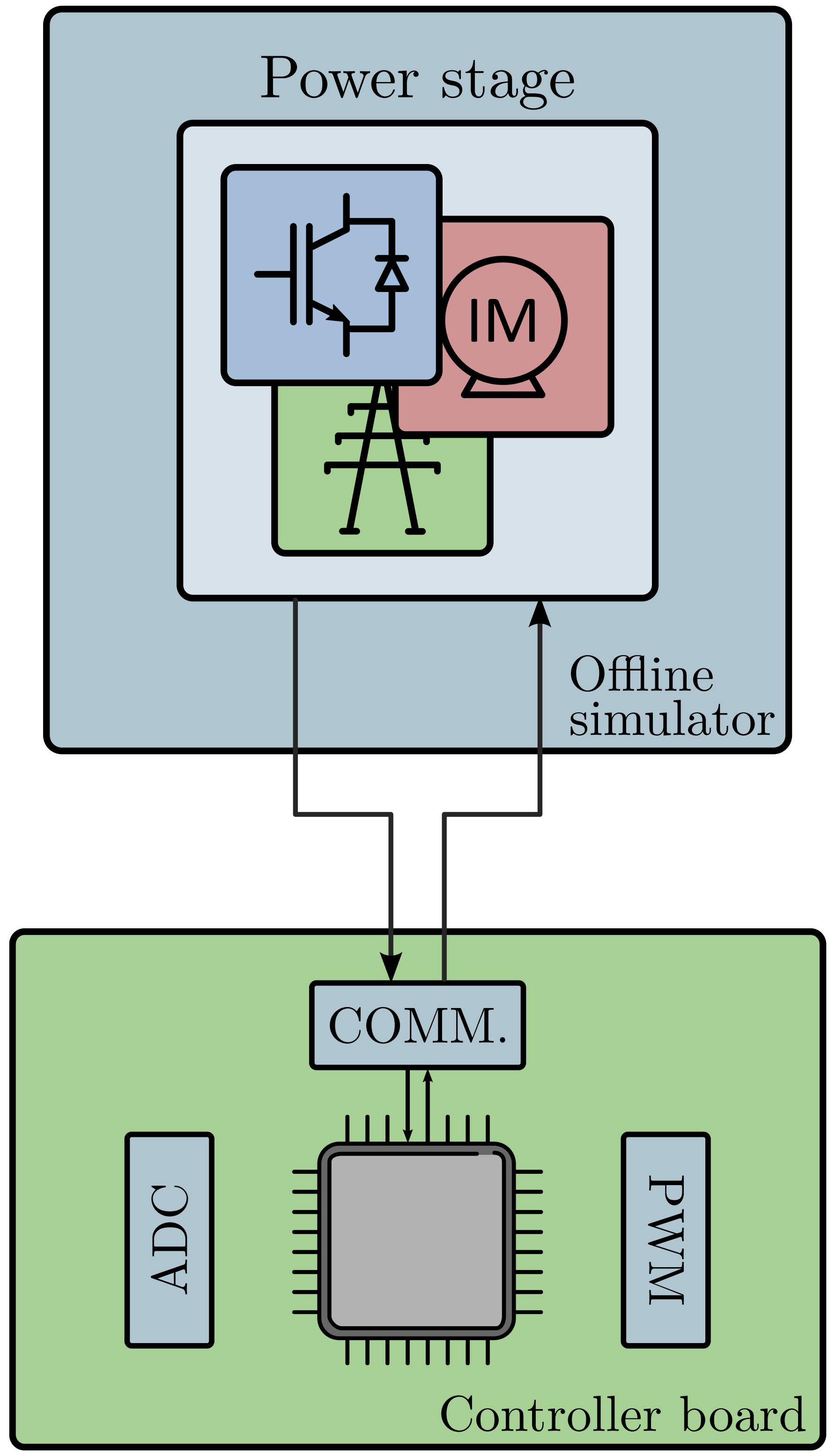
SIL



Processor in the loop

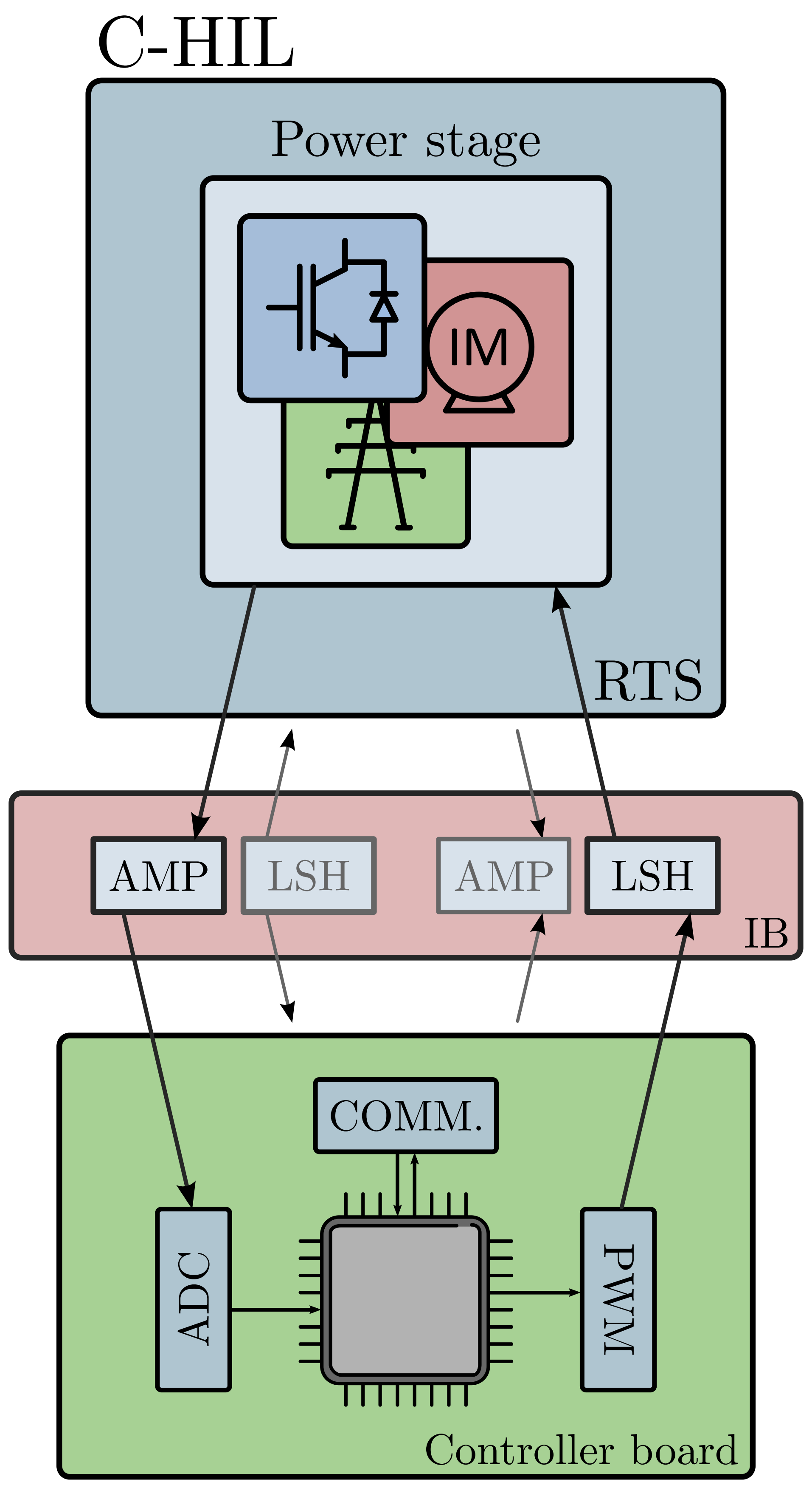
PC-based simulators «
Detailed modeling «
Safe & Slow «
Specific toolchain necessary «
Controller’s resources utilization known «

PIL



Real-time

Digital twinning

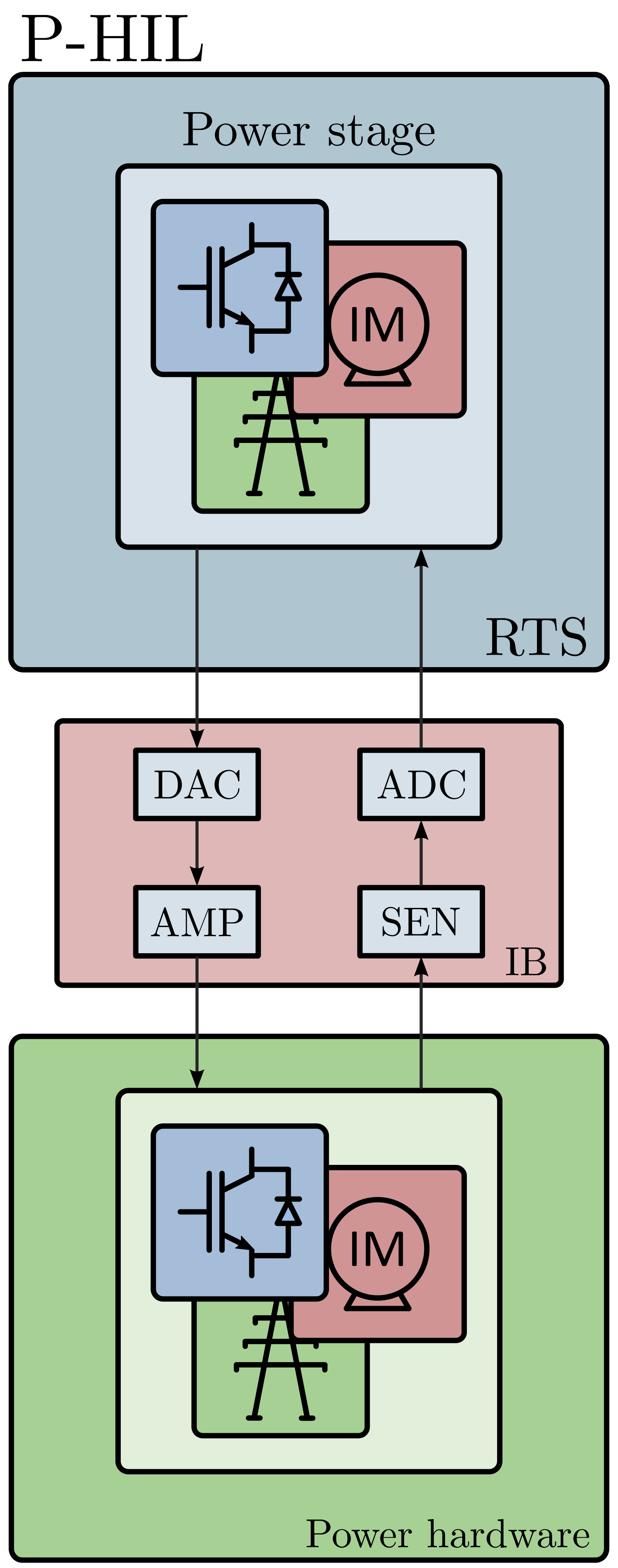


Controller hardware in the loop

» Dedicated hardware(emulators)-based
» Real-time execution (or faster than that!)
» Real controller and control code tested
» Safe & Fast
» It (also) enables test automation
» Simple emulator-controller interface

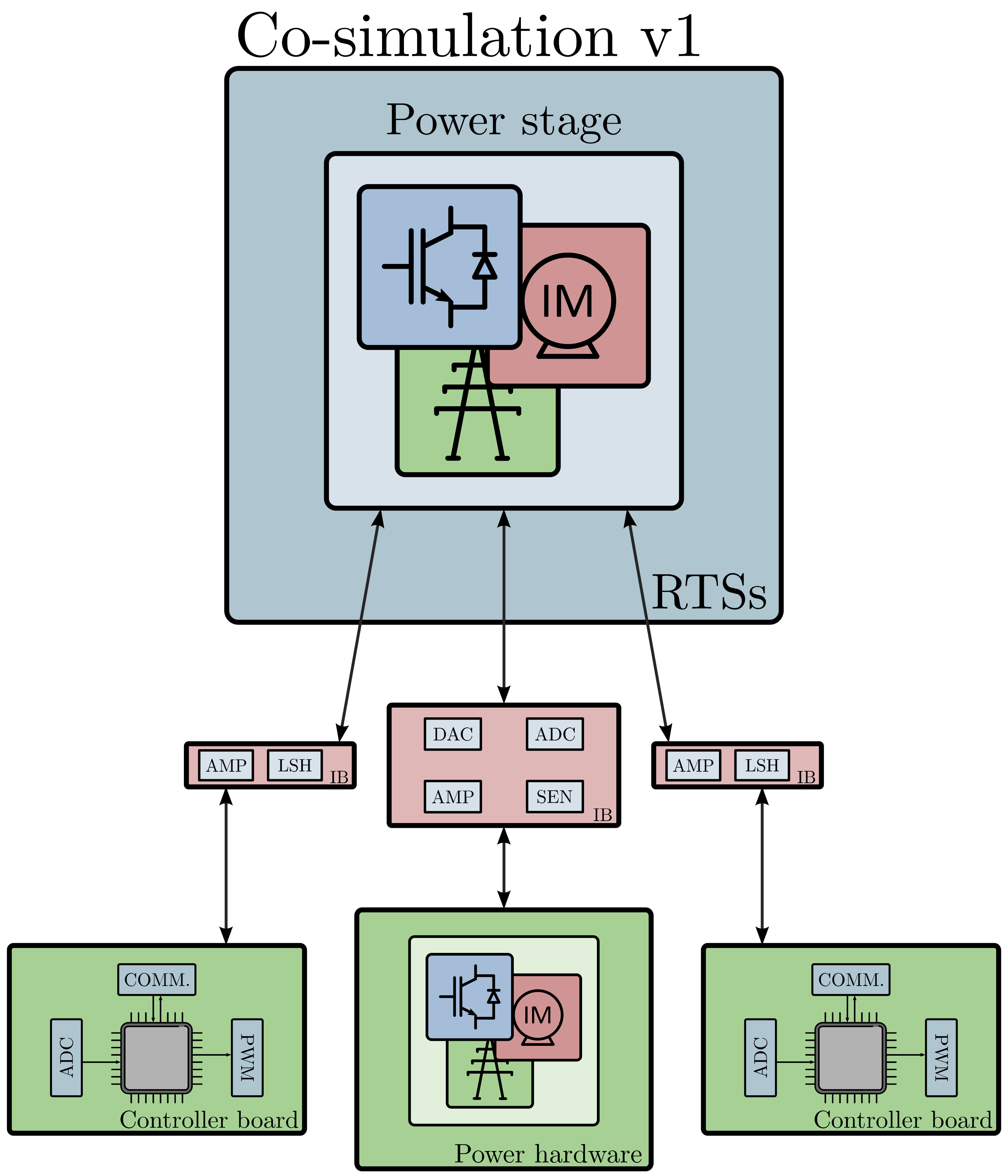
Power hardware in the loop

Dedicated hardware(emulators)-based «
Real-time execution «
Tests on real pieces of hardware «
Not so Safe & Fast «
Test automation difficult «
Complex interfaces necessary «



Co-simulation

Digital twinning



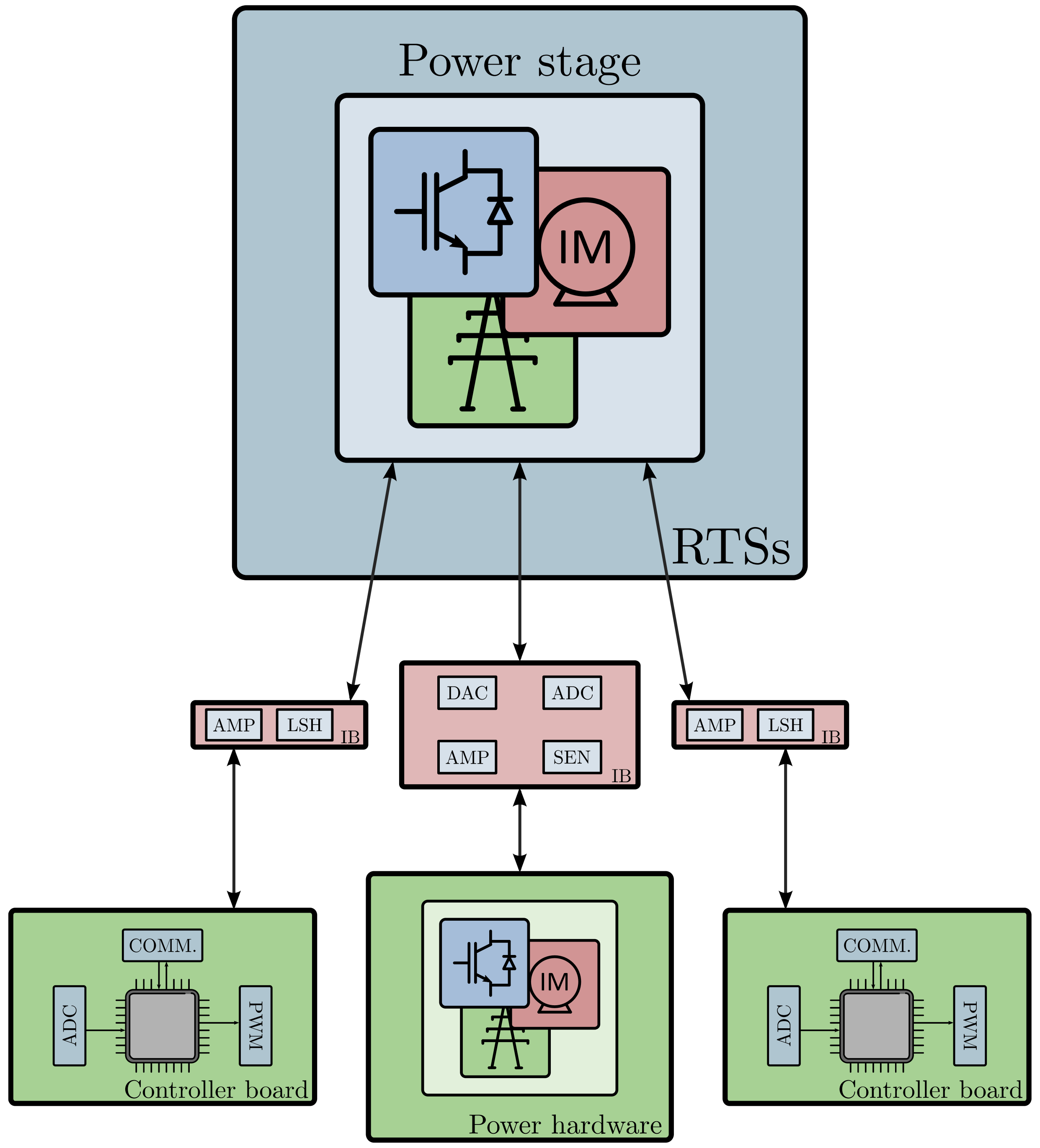
C-HILs + P-HILs

- Real-time execution
- Several (different) emulators used
- Real controller, control code and hardware equipment tested
- Complex
- Expensive
- Complex systems can be emulated and tested

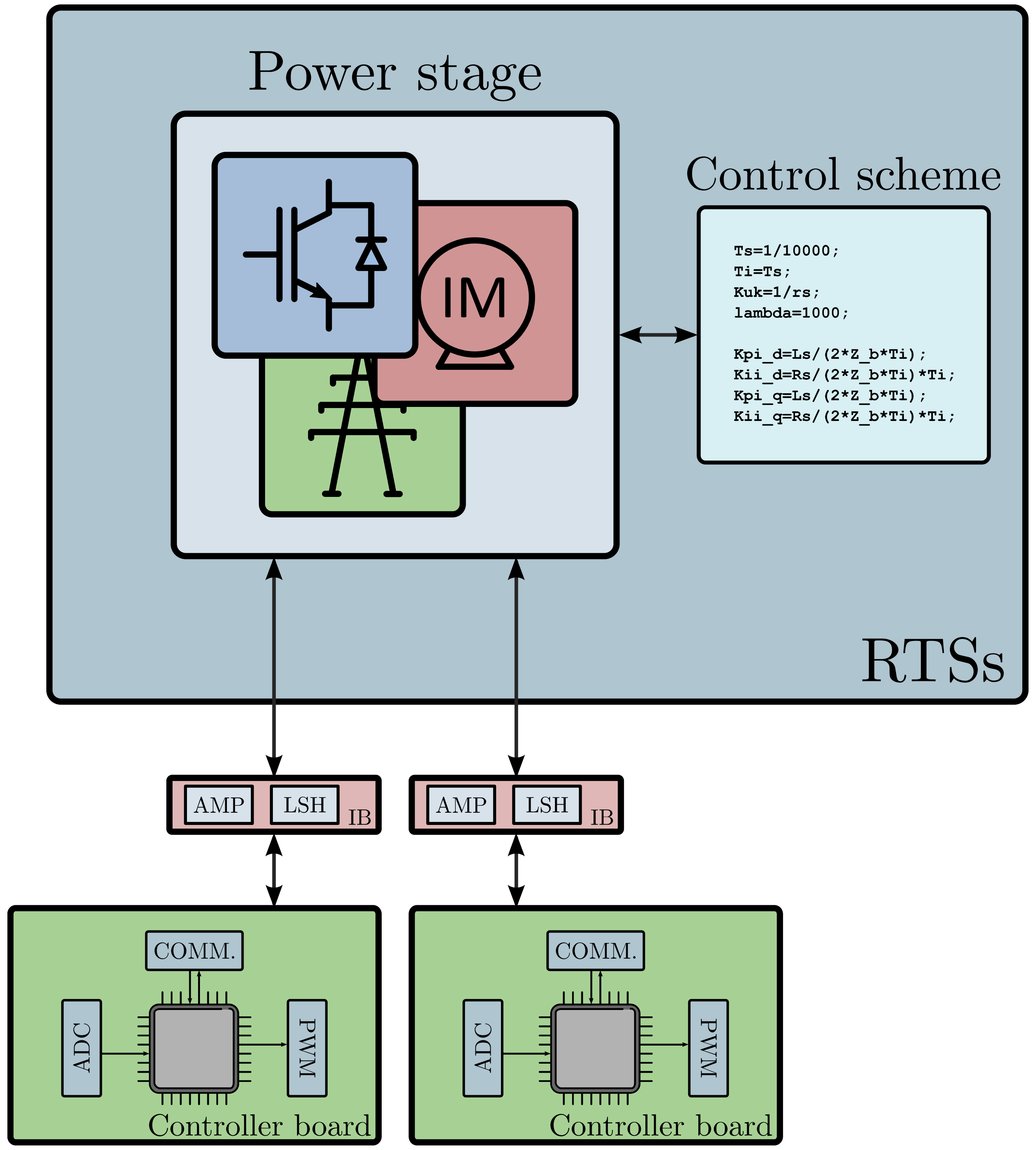
Co-simulation

Digital twinning

Co-simulation v1



Co-simulation v2.1



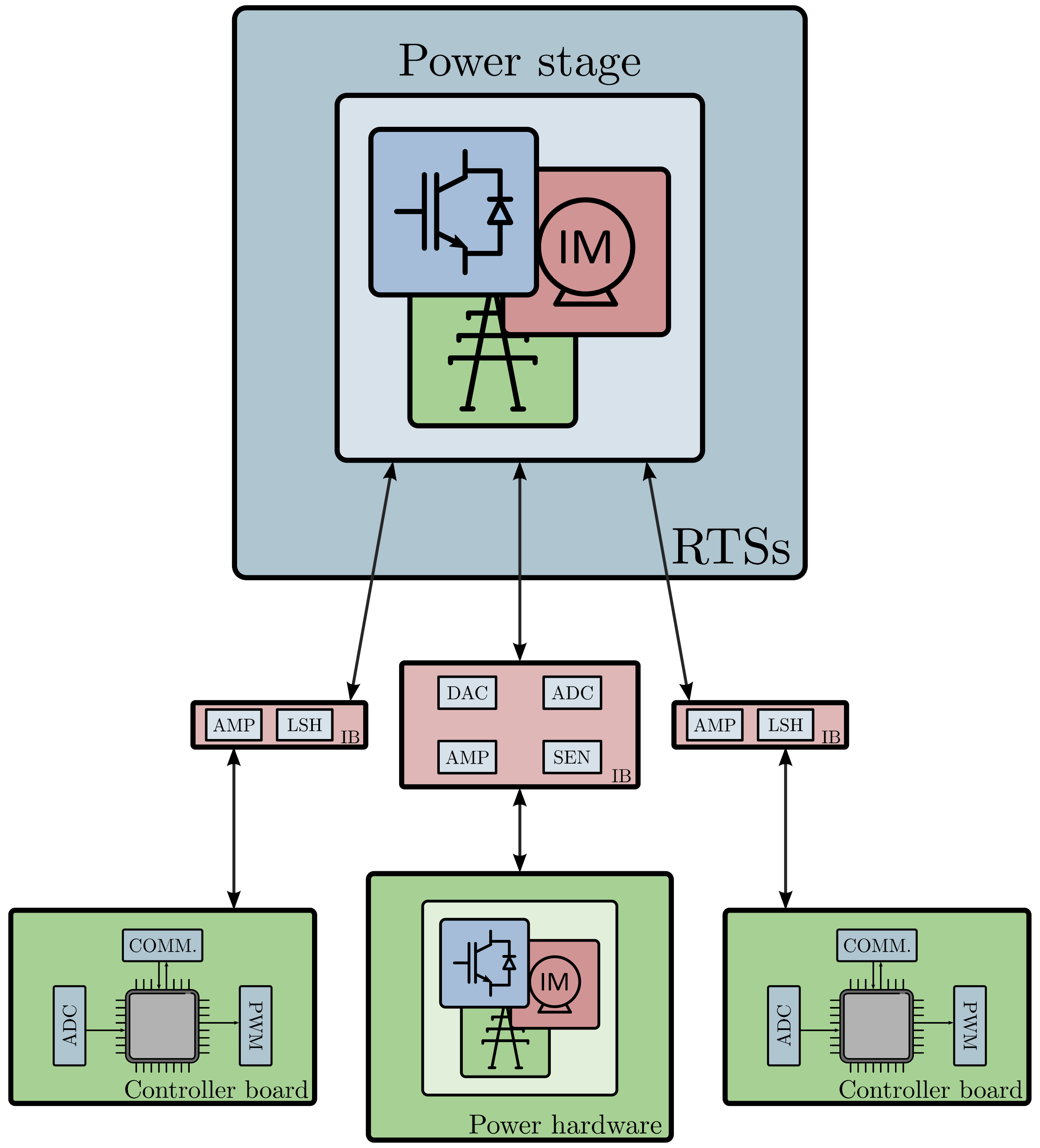
C-HILs

- Real-time execution
- Several (different) emulators used
- Real controller and control code tested
- Safe & Fast
- Complex systems can be emulated and tested

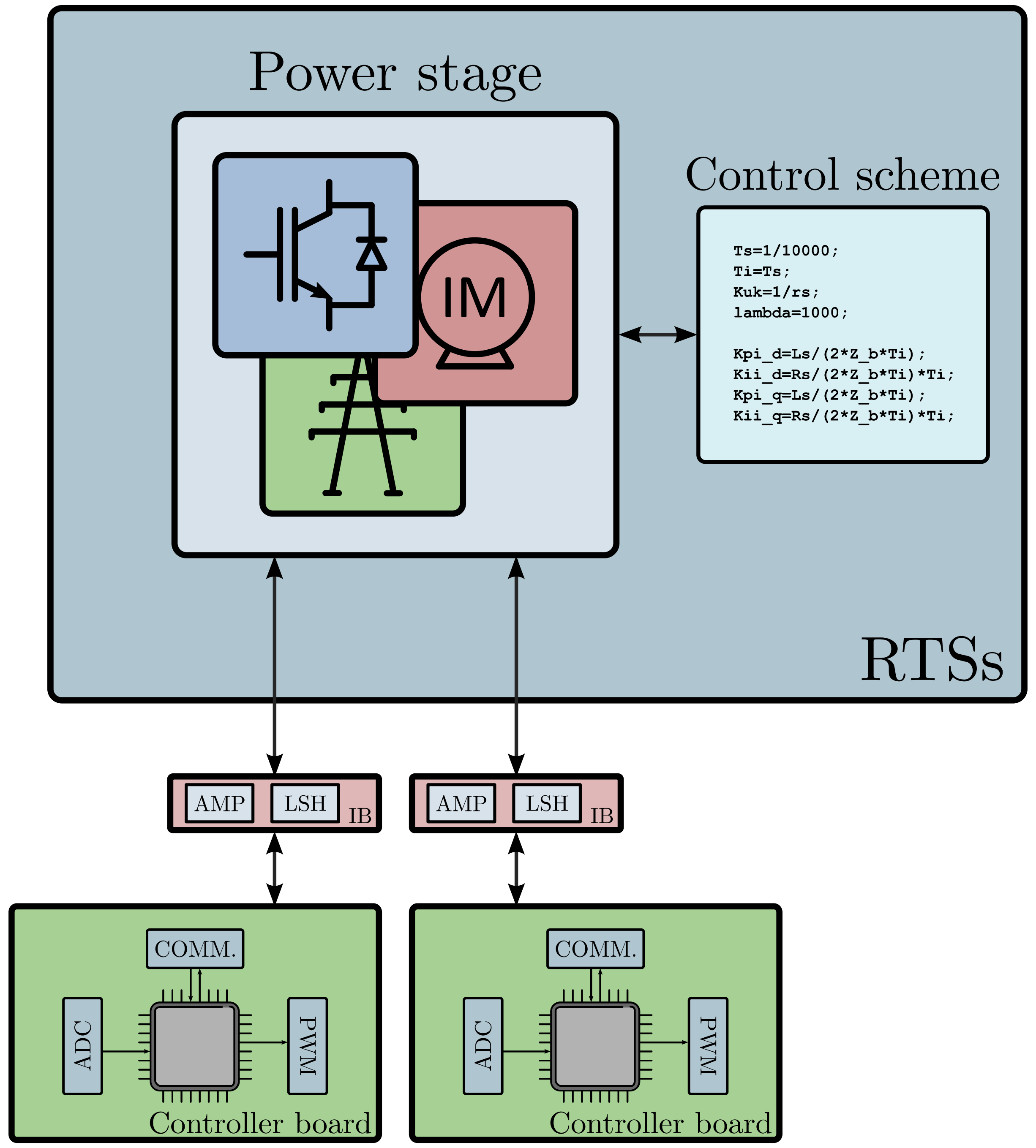
Co-simulation

Digital twinning

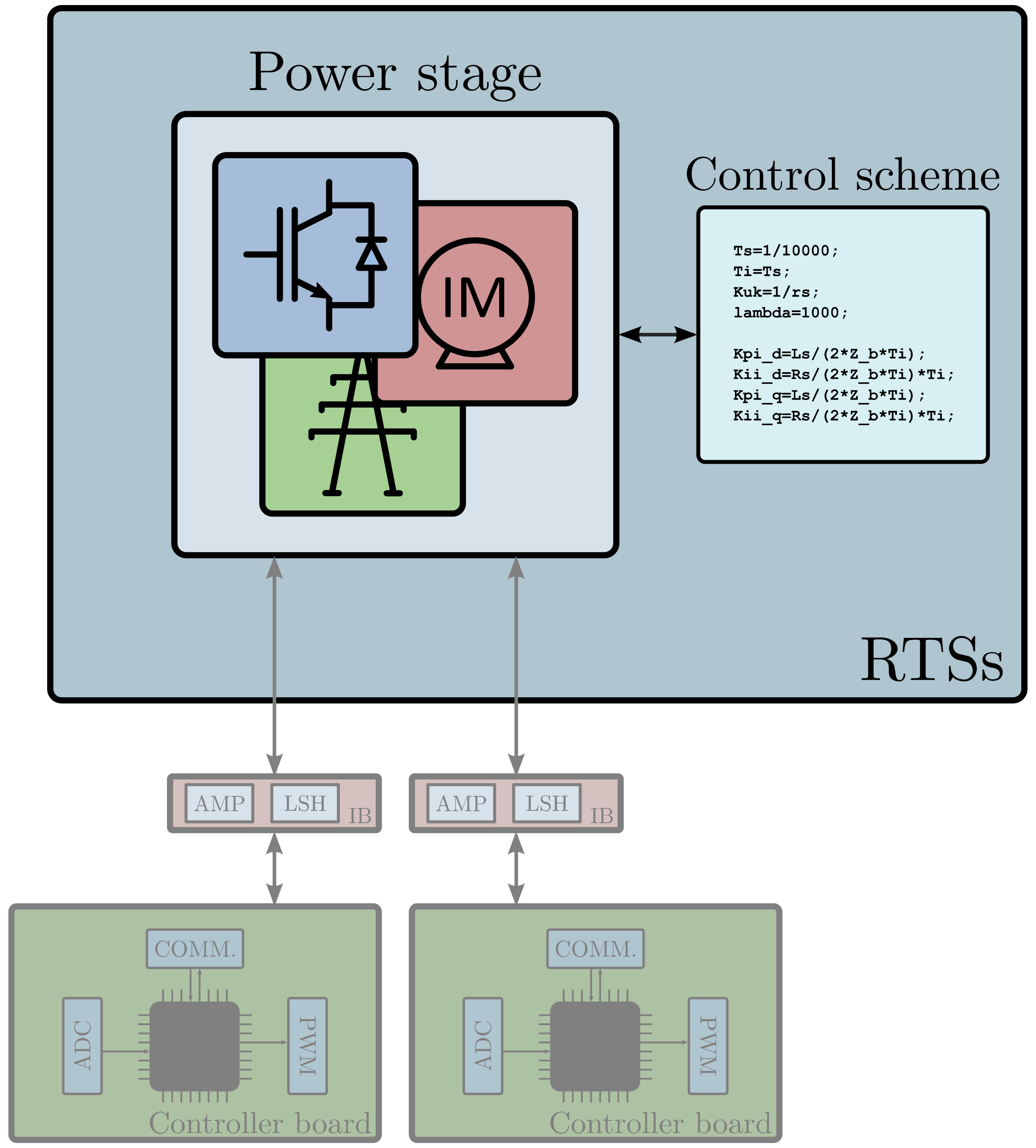
Co-simulation v1



Co-simulation v2.1



Co-simulation v2.2

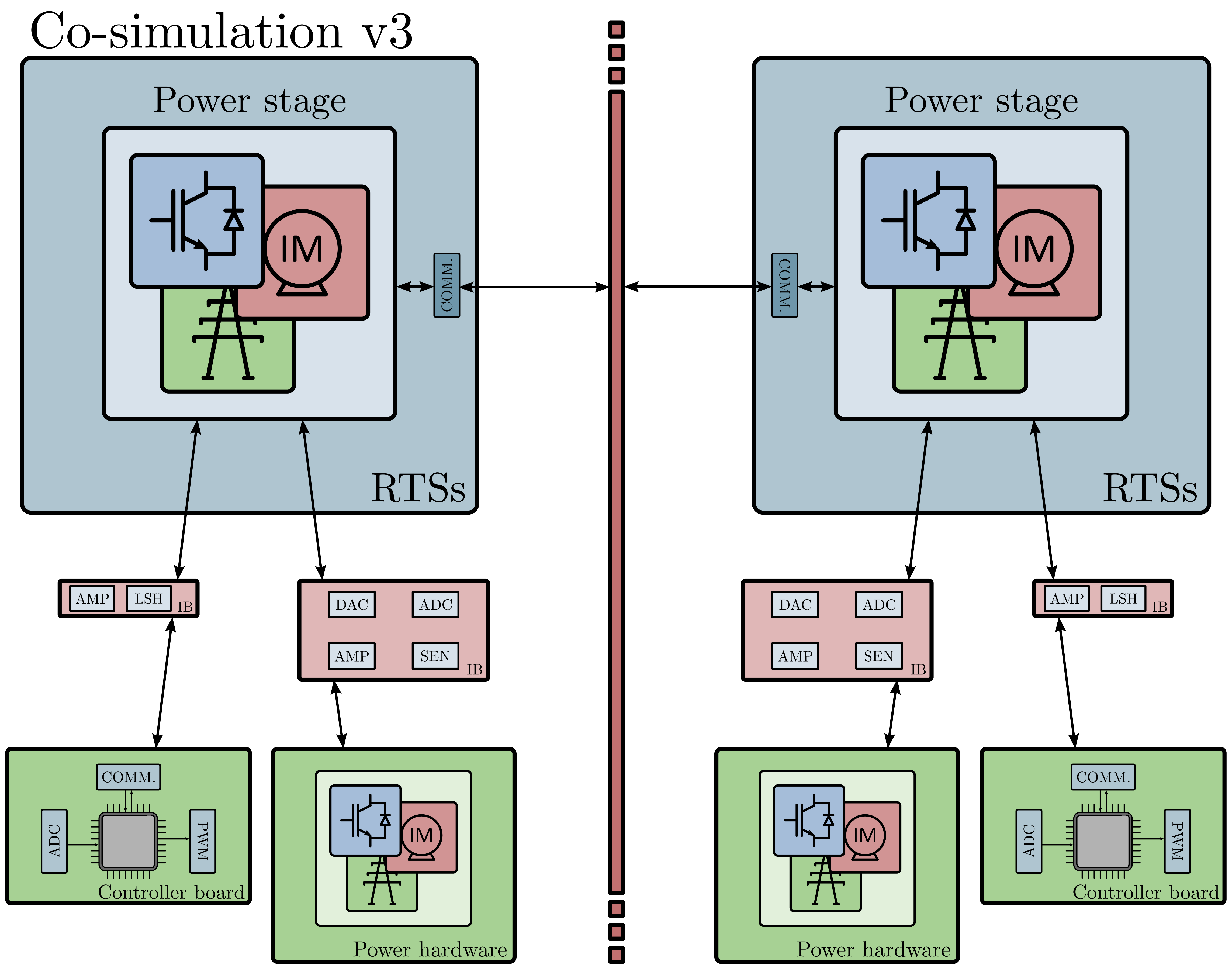


HILs

Some emulators can run both the emulated hardware and control schemes in real time (no external controller necessary) «
Faster than real-time execution (?) «
Complex systems can be emulated and tested «

Co-simulation

Digital twinning



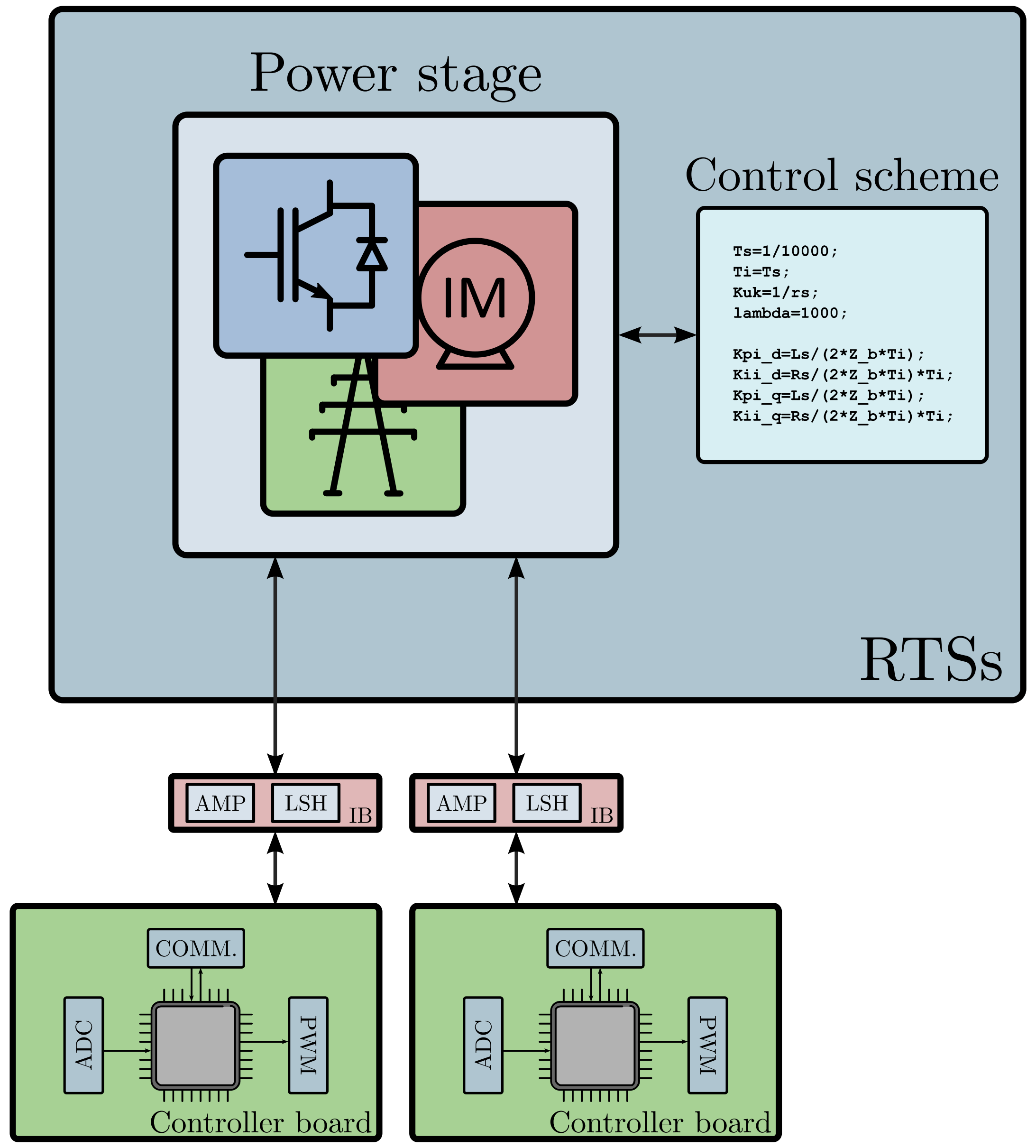
C-HILs + P-HILs (geographically dispersed)

» Several (different) emulators used
» Problem of latency
» Solves "problem" of privacy and data security
» Resources sharing
» Complex
» Difficult to manage
» Complex systems can be emulated and tested

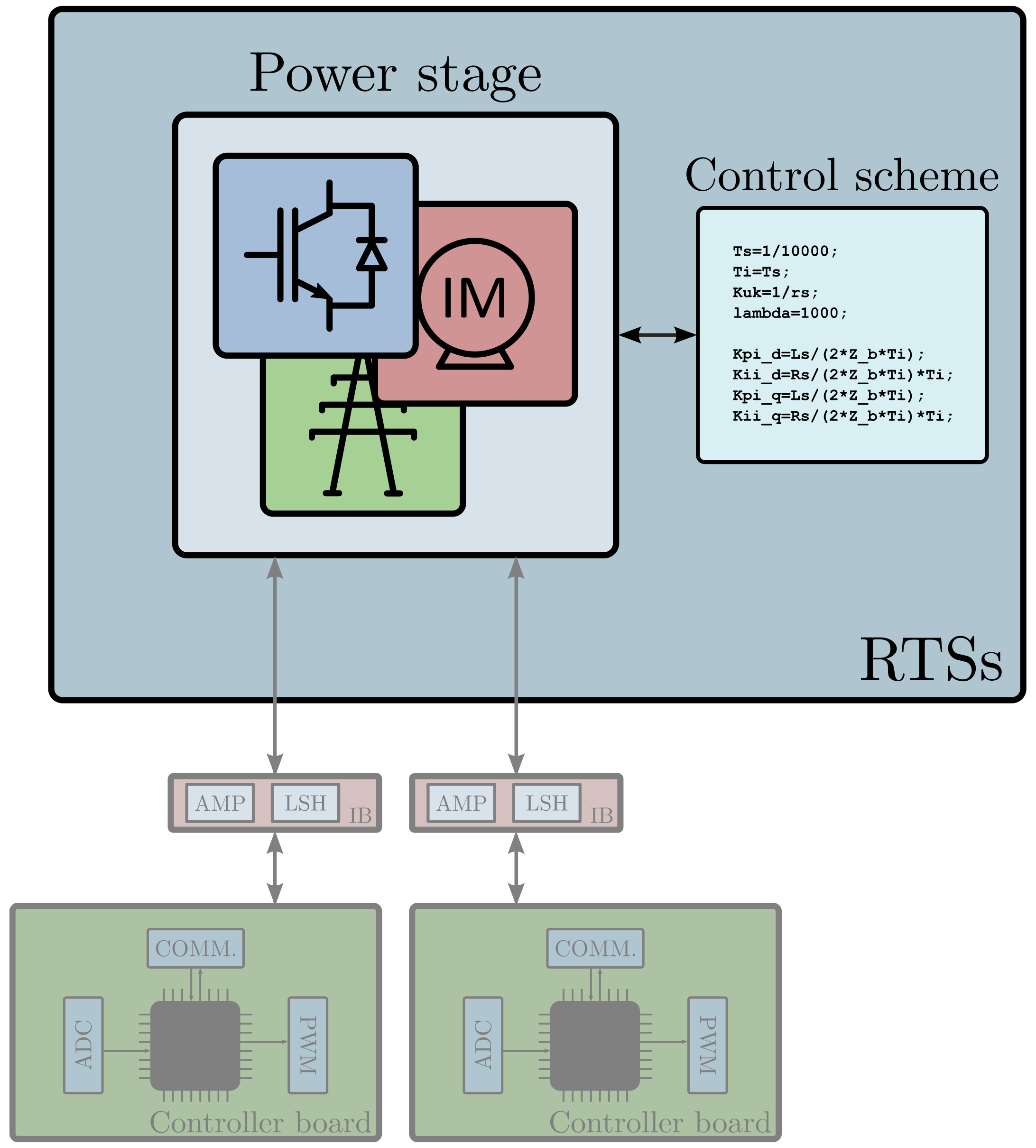
New trends

Digital twinning (relevant for CES)

Co-simulation v2.1

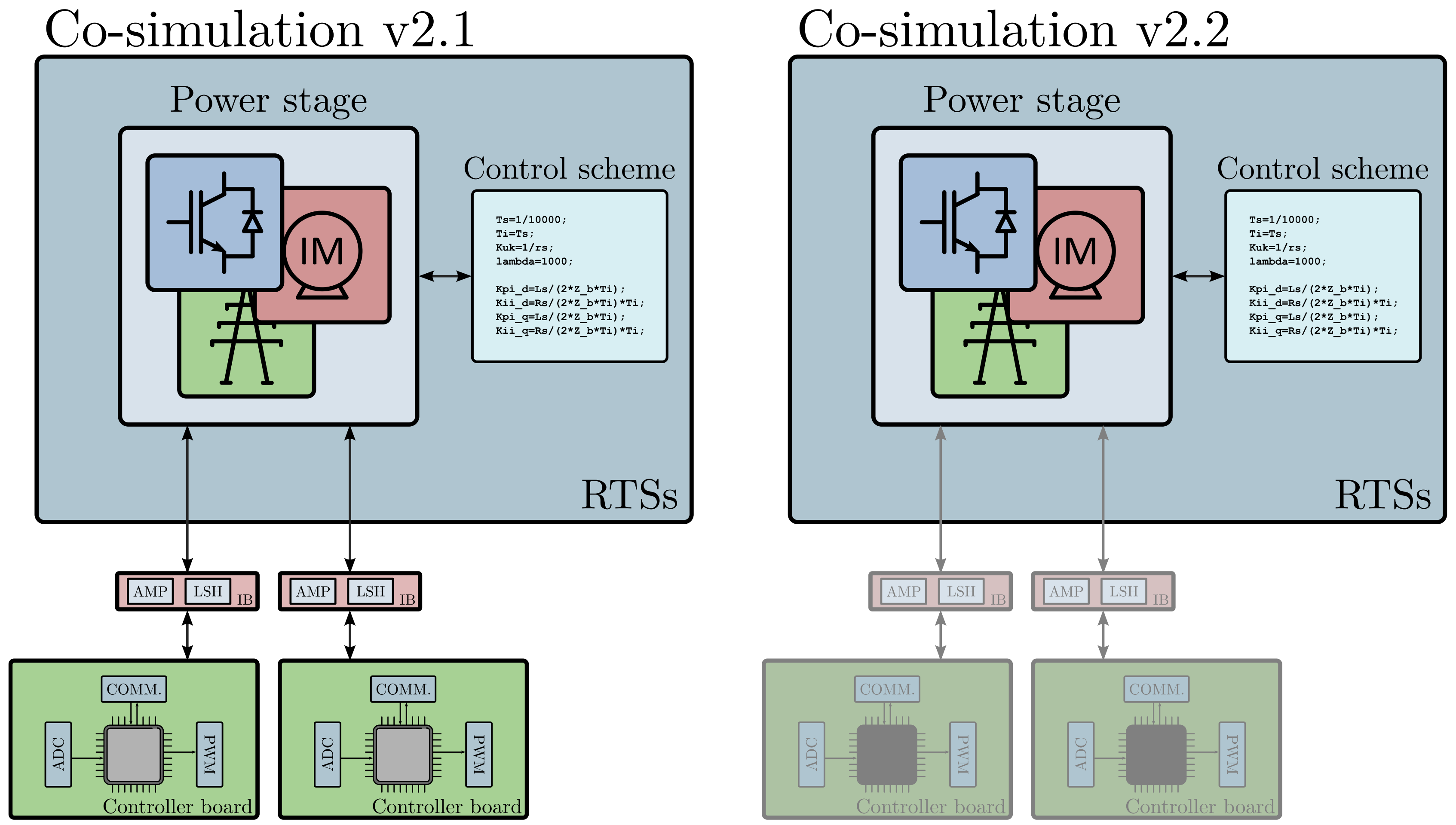


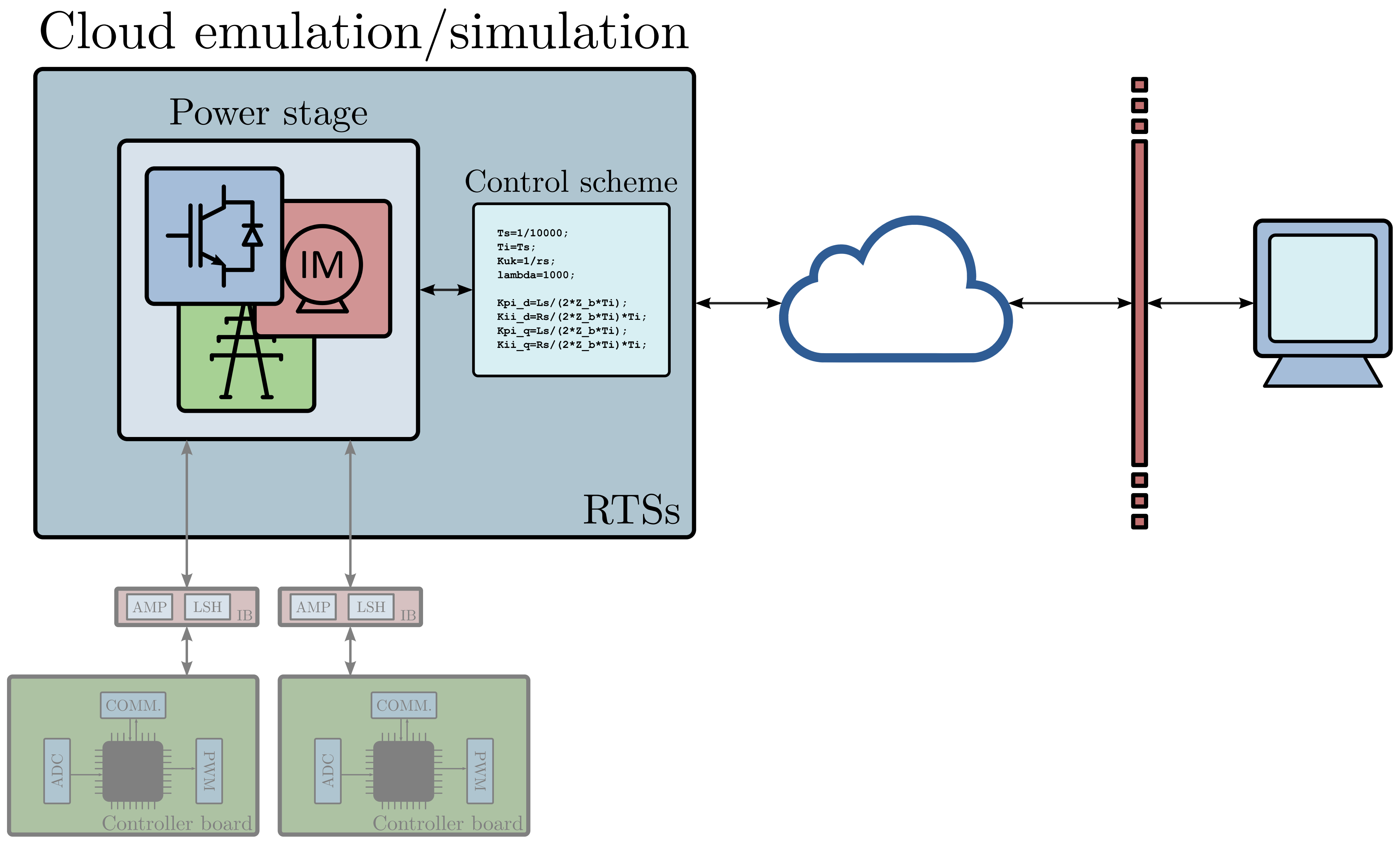
Co-simulation v2.2



New trends

Digital twinning (relevant for CES)



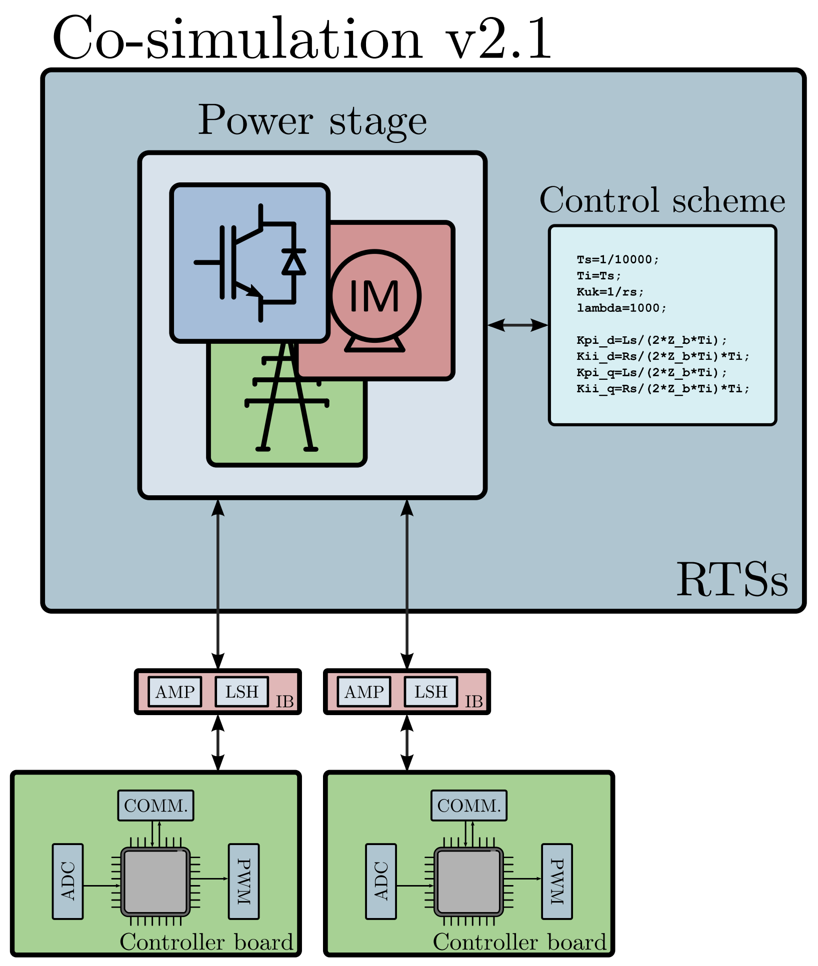


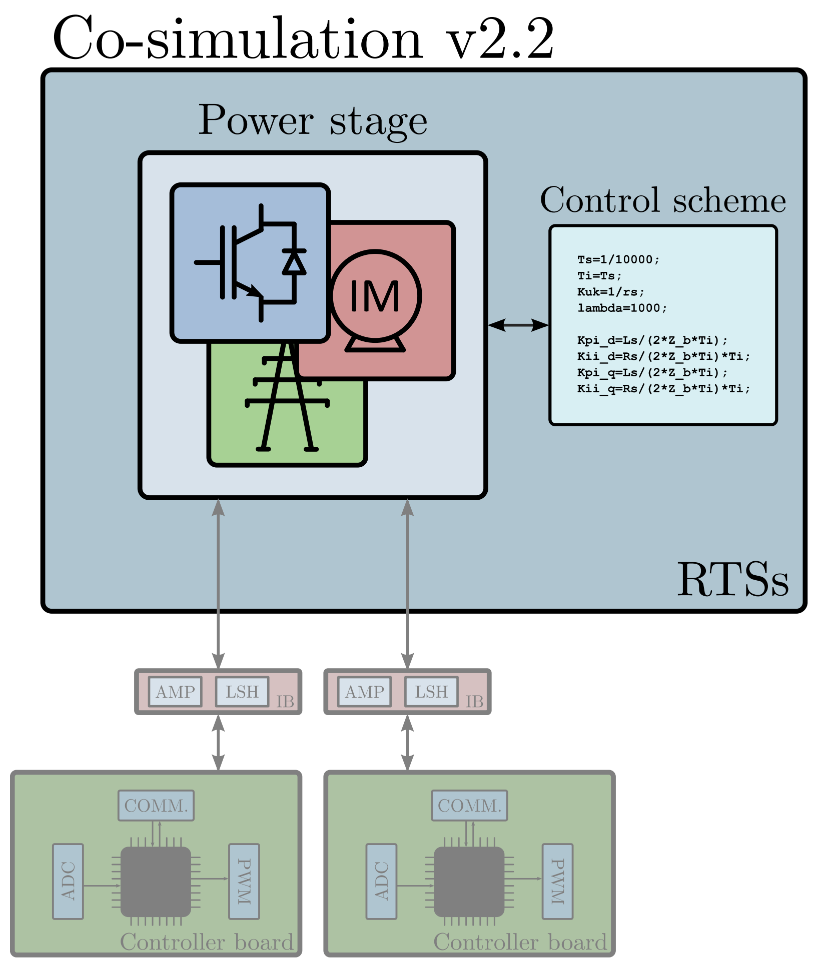
C-HILs or HIL behind the cloud

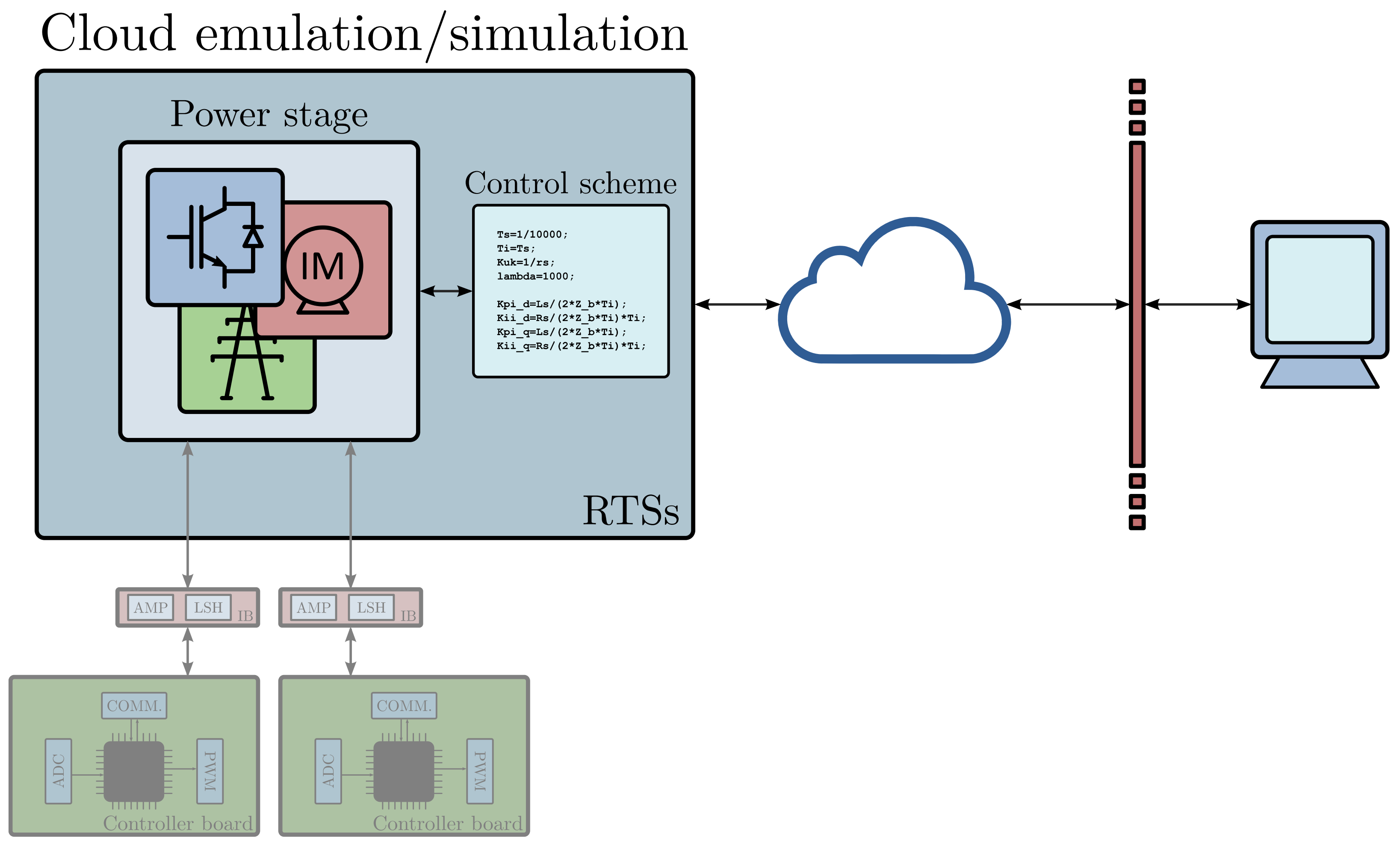
- Several (different) emulators can be used
- Real-time execution or slightly faster than real-time execution
- Enable activation of third-party services (models become part of a cloud ecosystem)
- Models creation and parametrization can be done via APIs (no ‘expert’ knowledge necessary)0

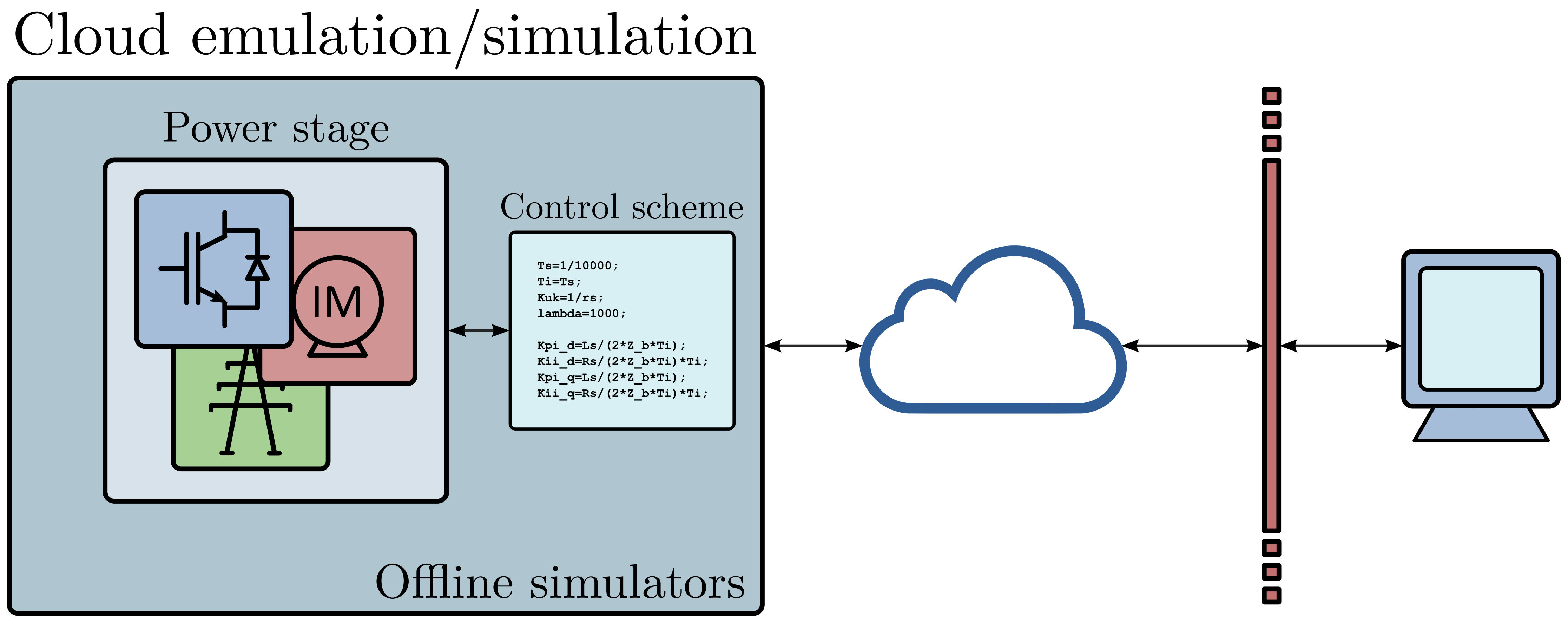
New trends

Digital twinning (relevant for CES)









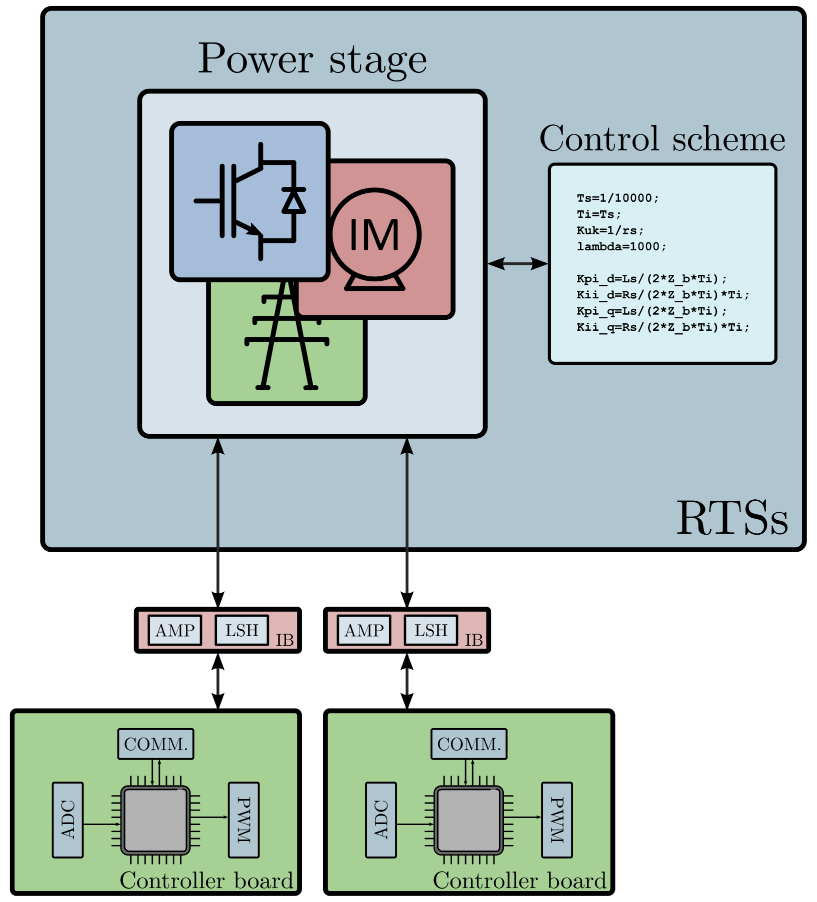
Offline simulator behind the cloud

- Enable activation of third-party services «
- Models creation and parametrization can be done via APIs «
- Significantly faster than real-time execution possible «
- Price – smaller precision (no transient phenomena modeled «
- Models ‘back/forward’ compatible «
- Change in resolution possible (simple) «

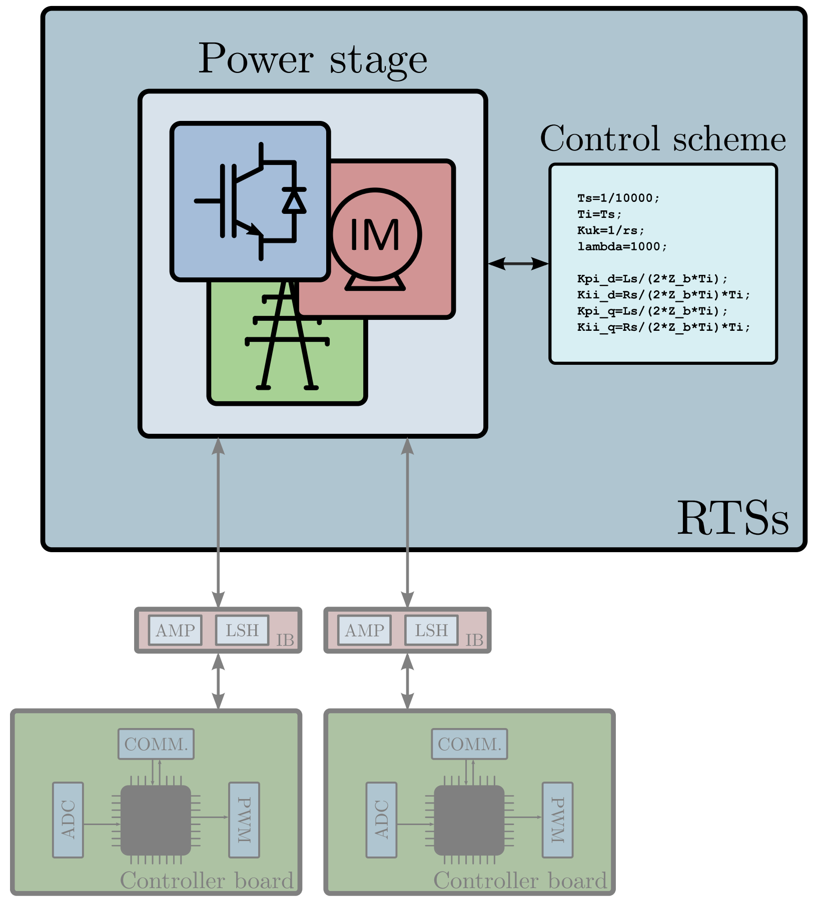
New trends

Digital twinning (relevant for CES)

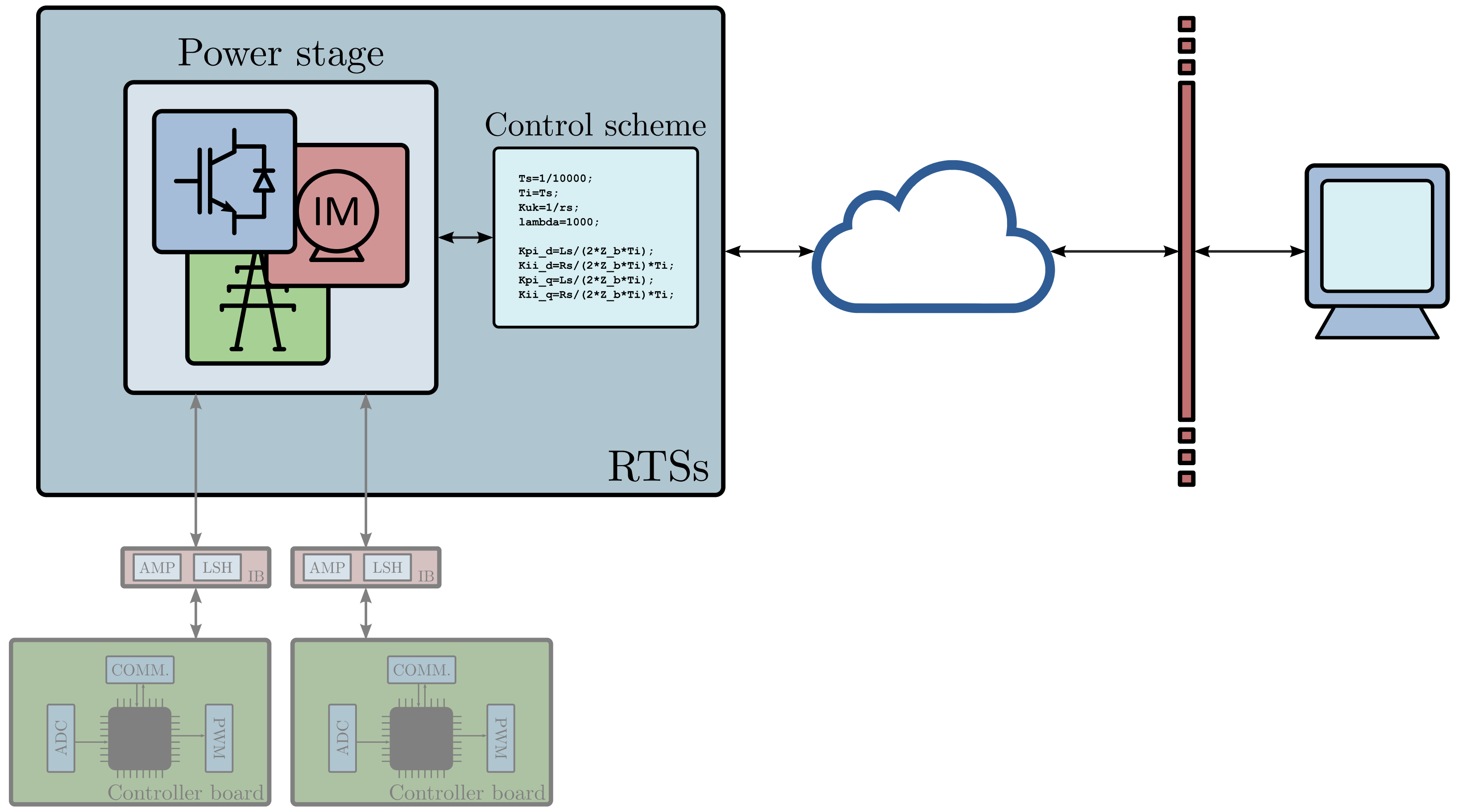
Co-simulation v2.1



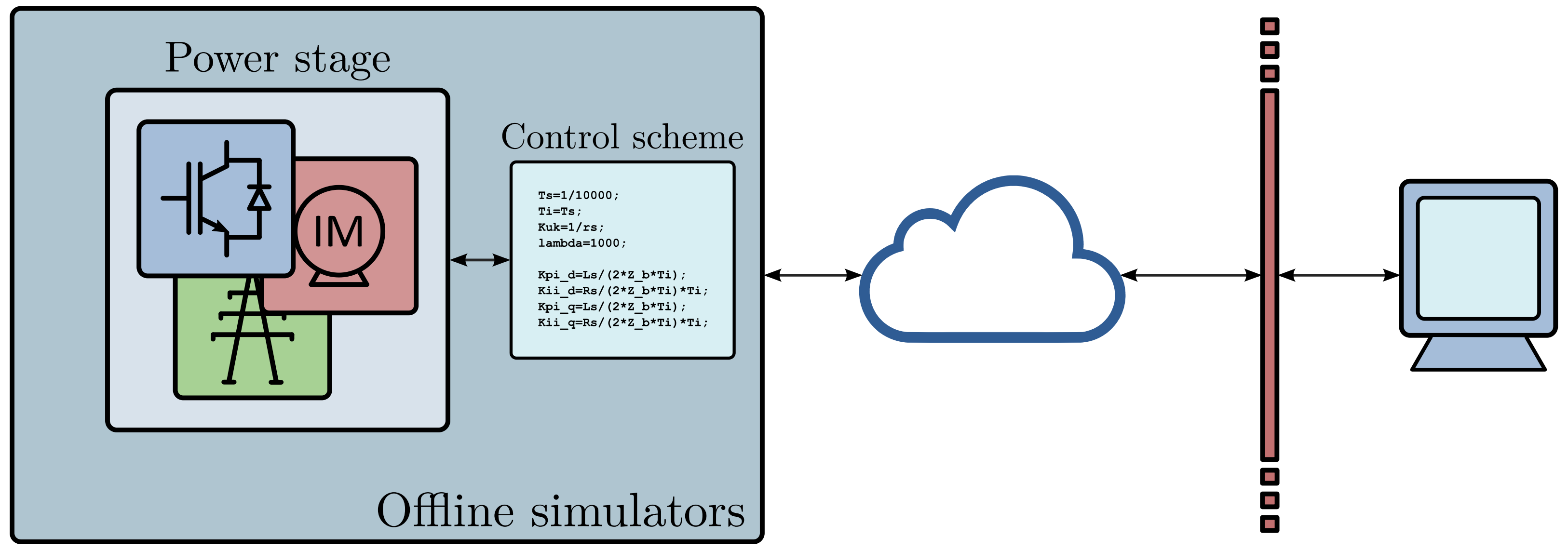
Co-simulation v2.2



Cloud emulation/simulation

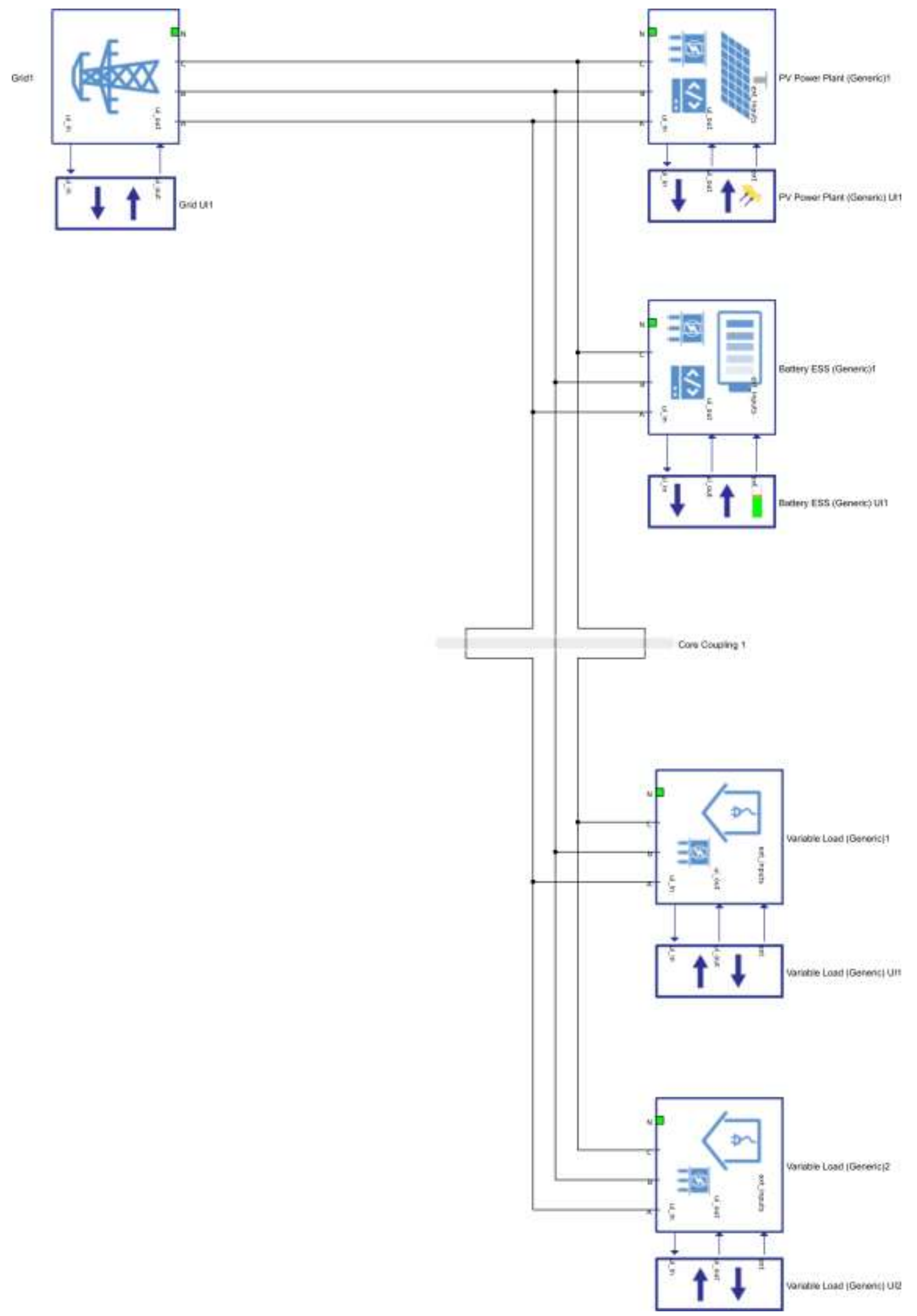


Cloud emulation/simulation

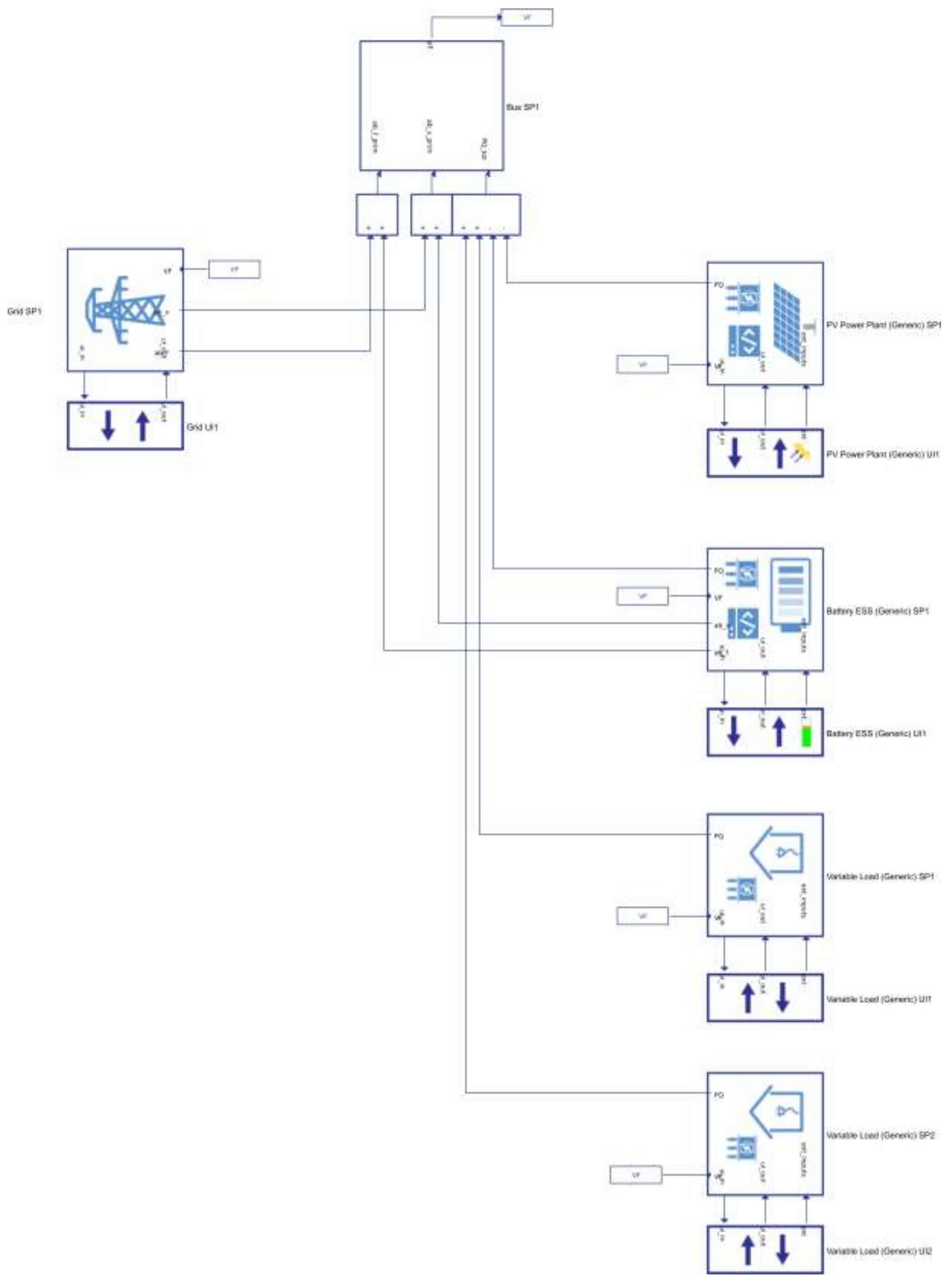


An example

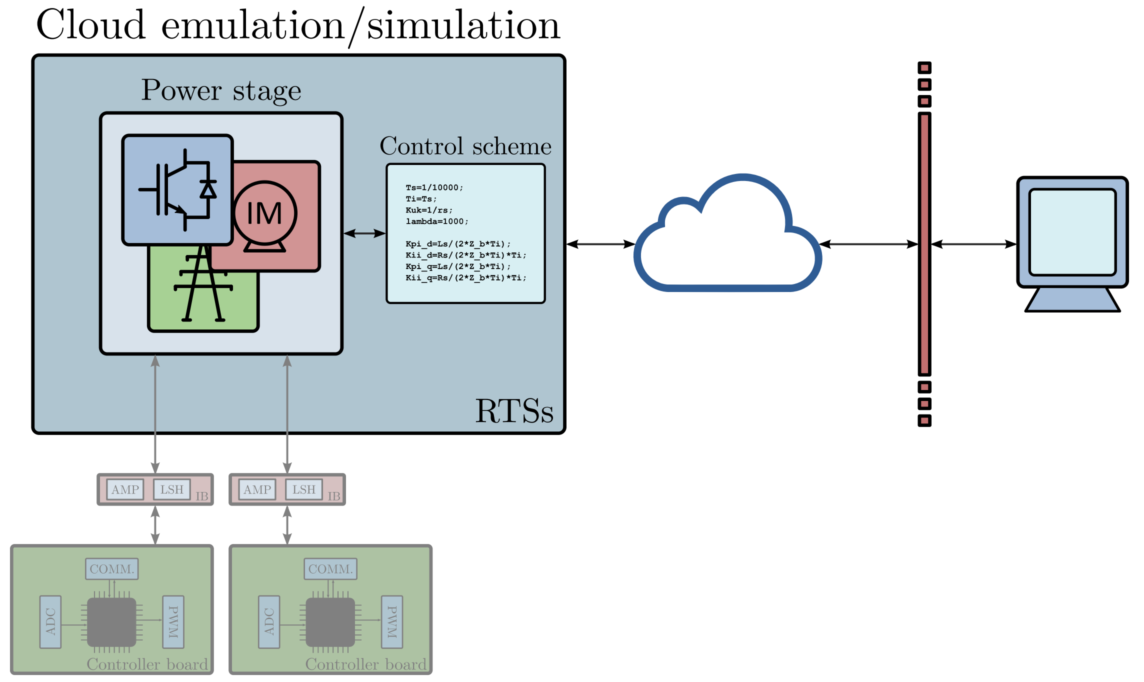
The model

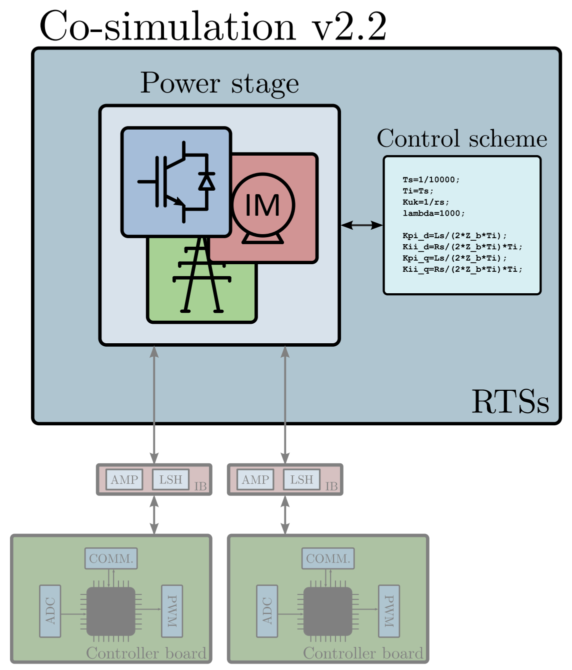


Approach 1



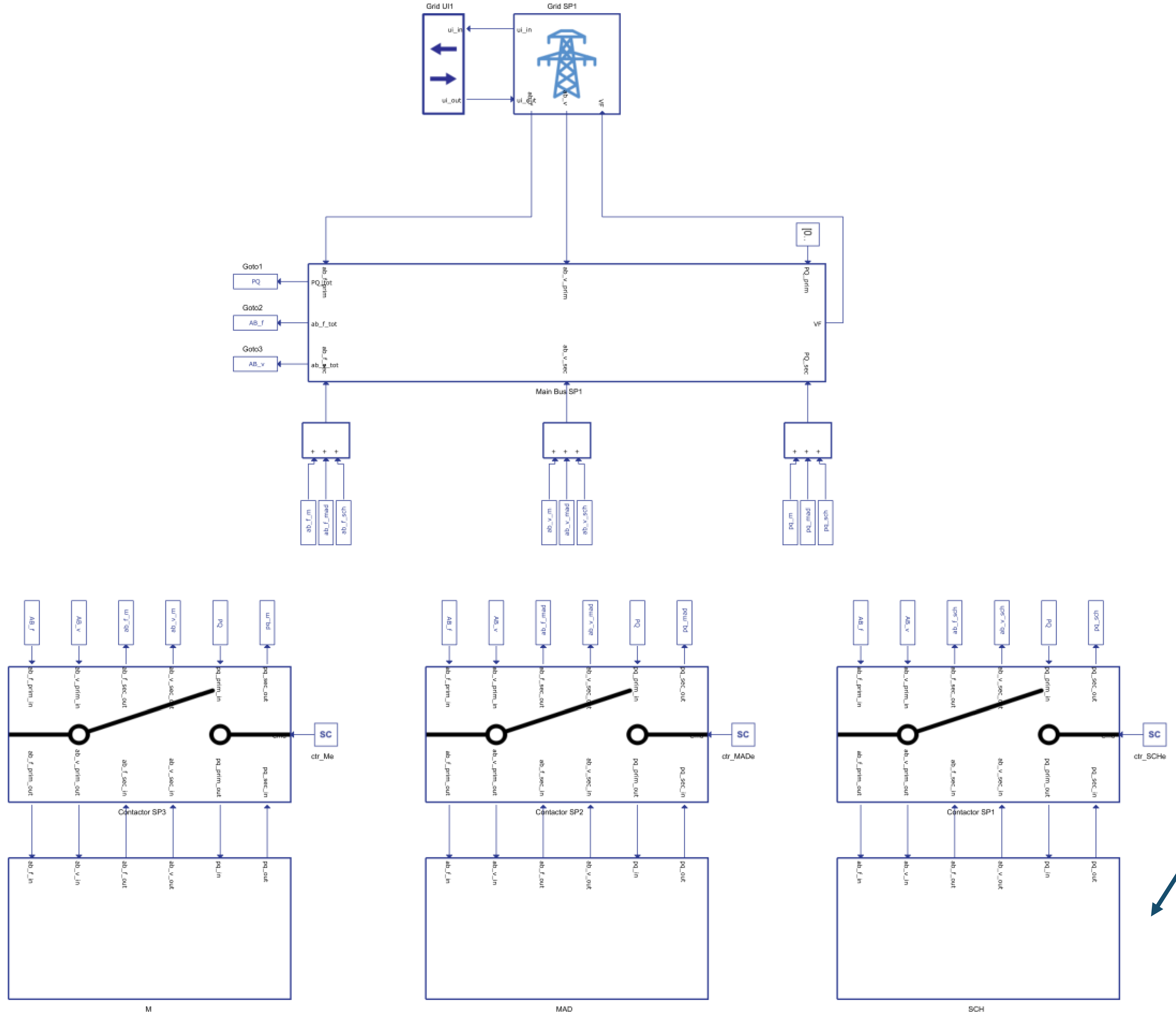
Approach 2

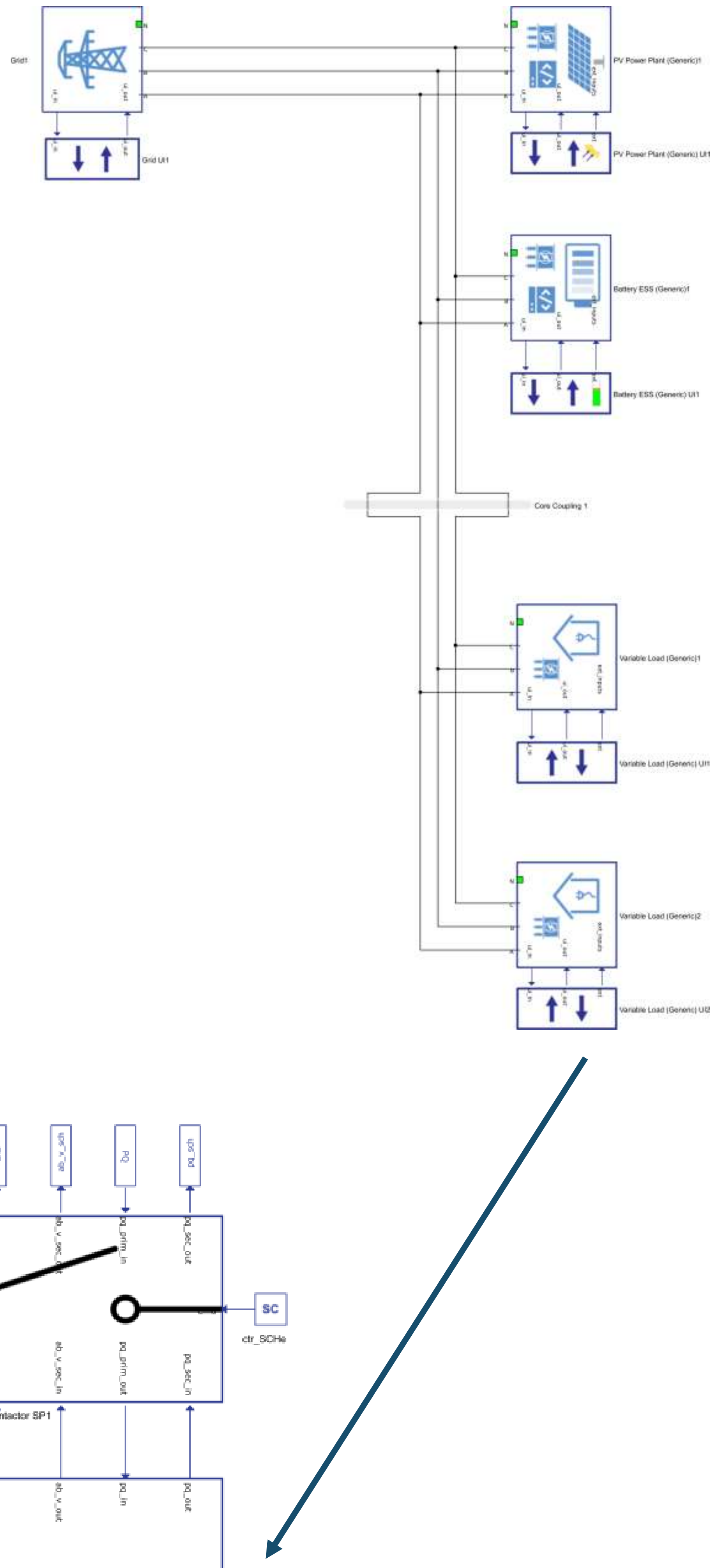


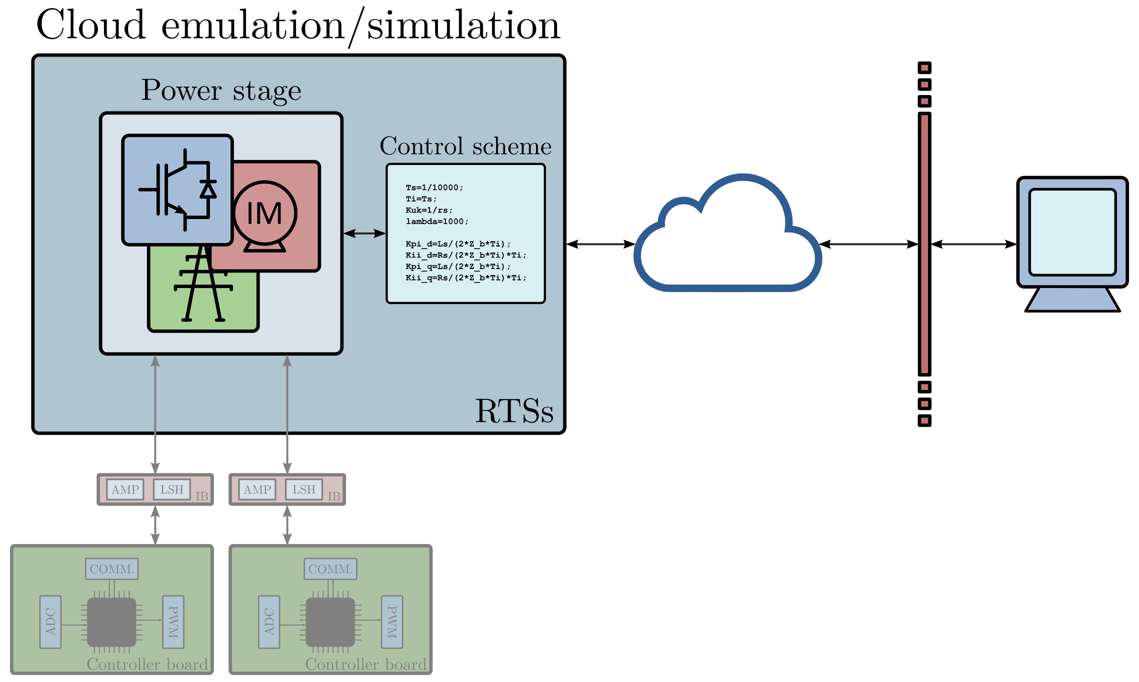


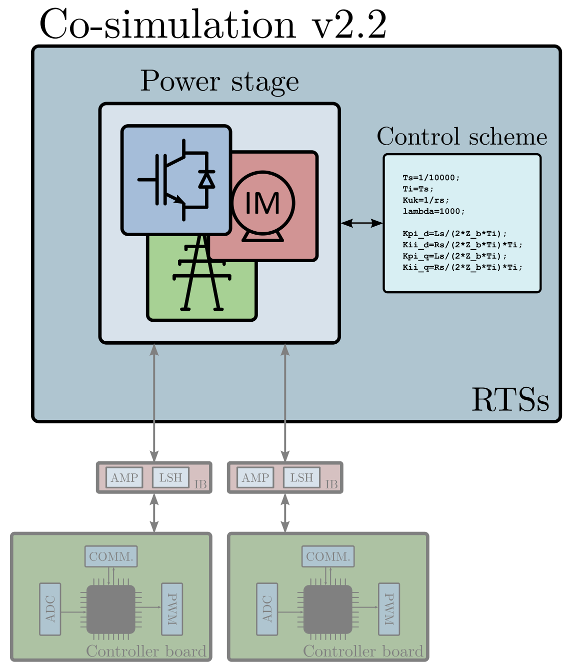
An example

The model









3 PV systems
3 Battery systems
6 Passive loads nodes

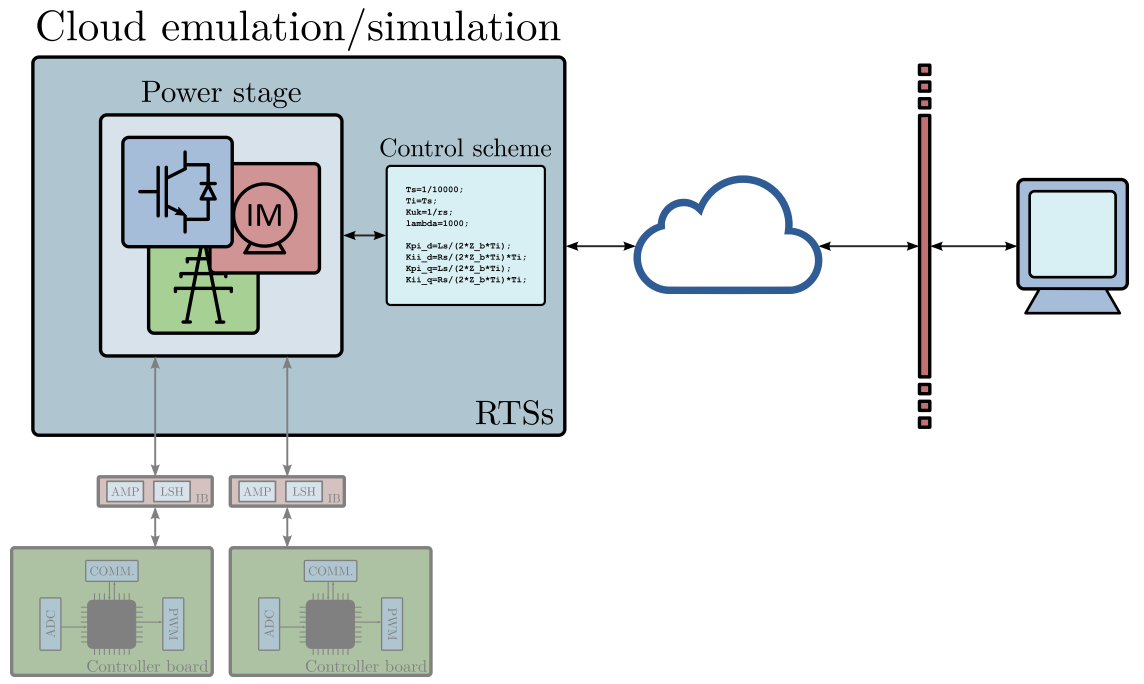
An example

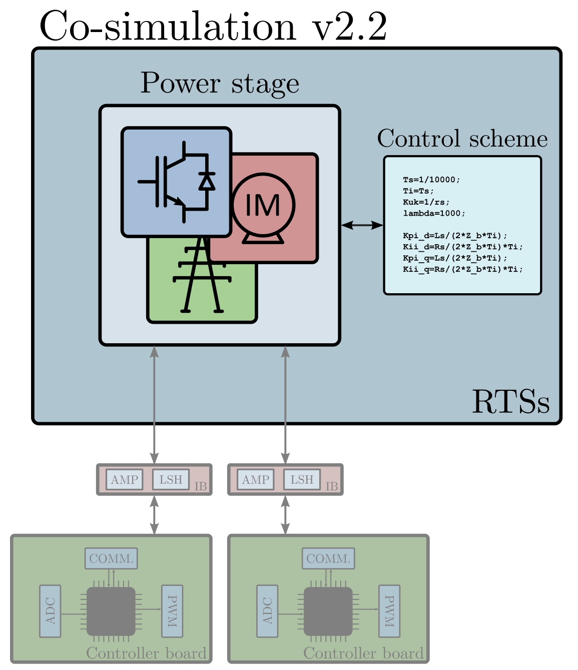
Runtime data

Models can be executed:
- » On emulators
- » On local PC (offline simulation)
- » In/through the cloud (servers)

On emulators, models can:
- » Run in real-time (approach 1)
- » Run slightly faster than in real-time (approach 1)
- » Run much faster than real-time (approach 2)

On local PC, models can be:
- » ~Slower than real-time (approach 1)
- » Faster than real-time (approach 2)





An example

Runtime data

Models can be executed:

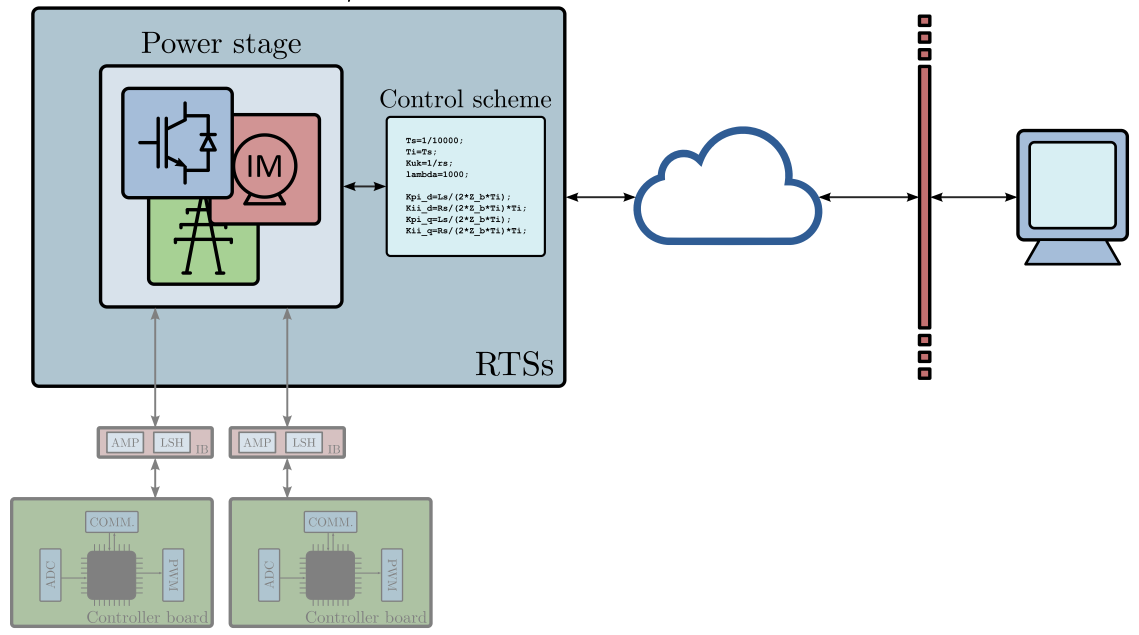
» On emulators
» On local PC (offline simulation)
» In/through the cloud (servers)

Execution on emulators (HIL 602+):

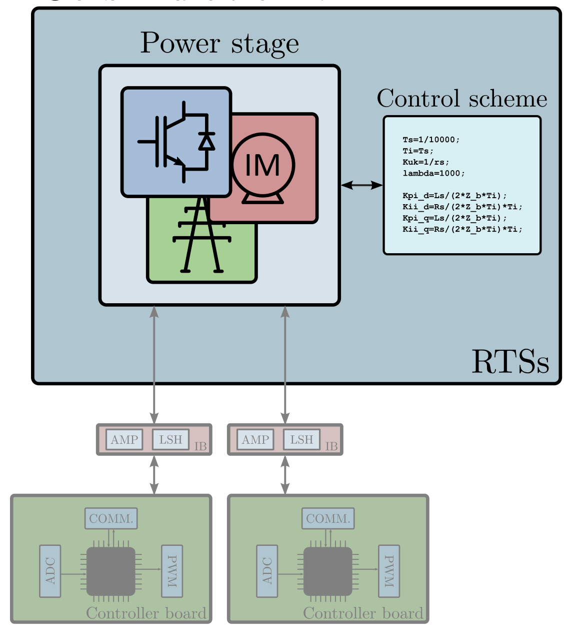
Approach 1		Config. 1	Config. 2
Time step	t=1e-6	RT	RT
	t=5e-6	RT	RT
	t=20e-6	unstable	unstable

Approach 2		Config. 1	Config. 2
Time step	t=1e-6	25/100	23/100
	t=5e-6	5/100	3/100
	t=20e-6	1/100	1/100

Cloud emulation/simulation



Co-simulation v2.2



An example

Runtime data

Models can be executed:
- » On emulators
- » On local PC (offline simulation)
- » In/through the cloud (servers)

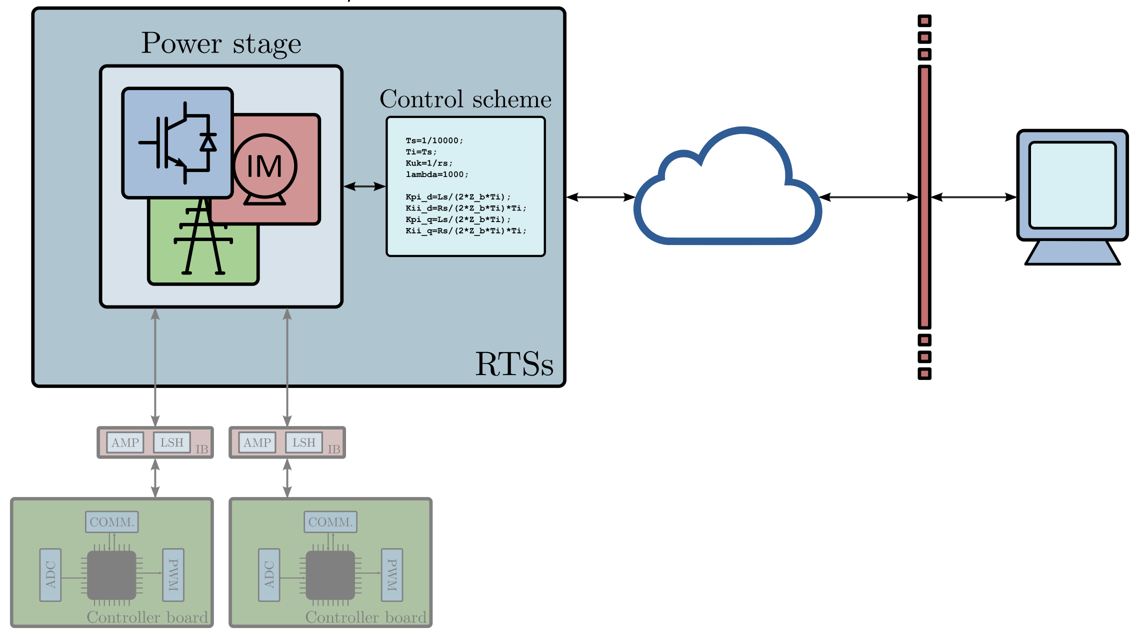
Execution on local PCs:

Approach 1		Config. 1	Config. 2
Time step	t=1e-6	122/100	120/100
	t=5e-6	90/100	90/100
	t=20e-6	unstable	unstable

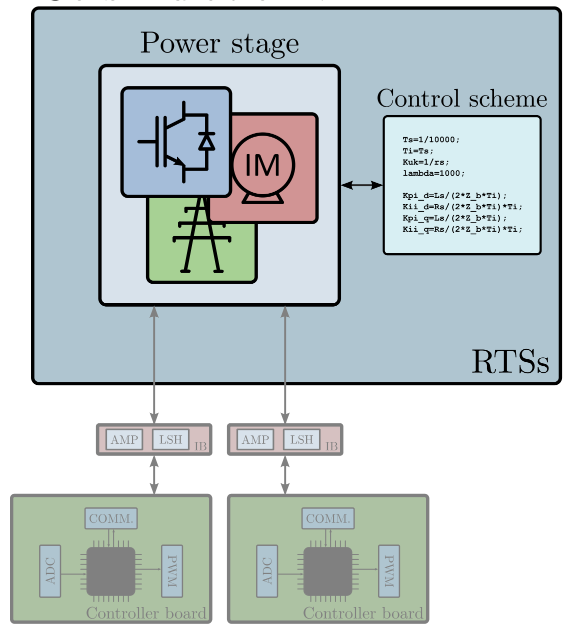
Approach 2		Config. 1	Config. 2
Time step	t=1e-6	42/100	41/100
	t=5e-6	11.5/100	10/100
	t=20e-6	4.5/100	4/100

	Config. 1	Config. 2
Processor	Intel(R) i5-8250U CPU @ 1.60GHz	Intel(R) i7-4790K CPU @ 4.00GHz
Video Card	Intel(R) UHD Graphics 620	Microsoft Basic Display Adapter
RAM	8.0 GB	8.0 GB
OS	Windows 10	Windows 10

Cloud emulation/simulation



Co-simulation v2.2

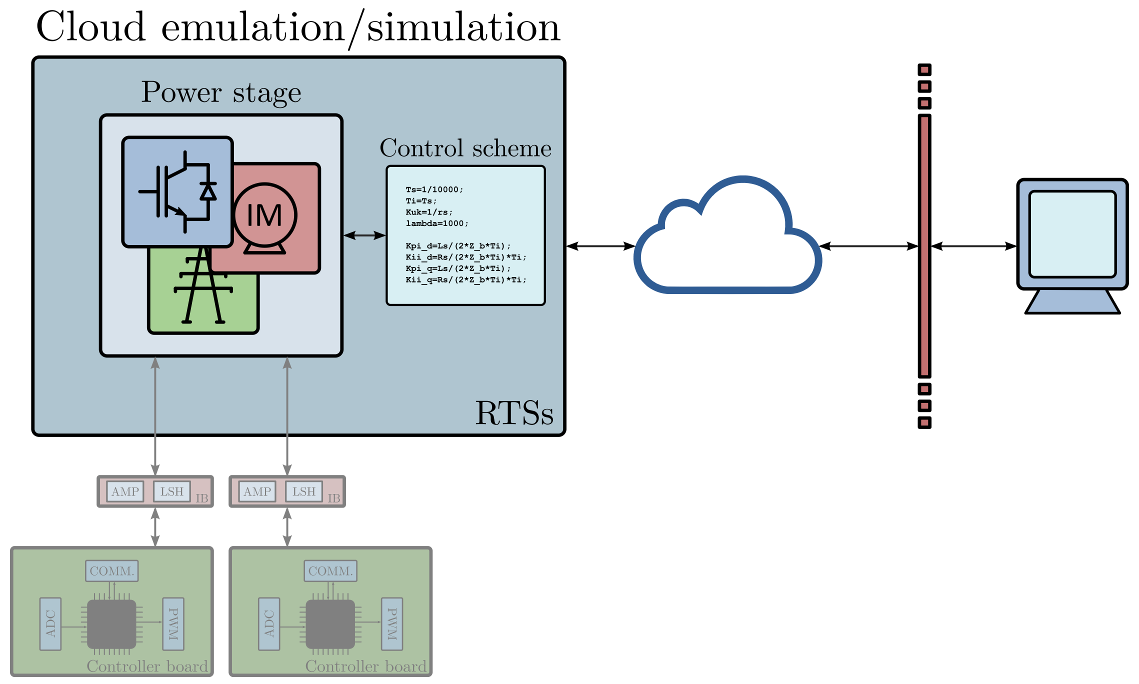


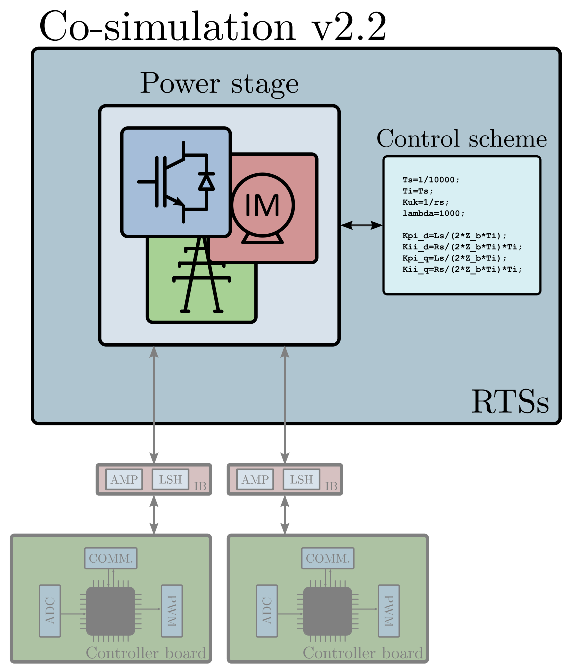
An example

Runtime data

Models can be executed:
- On emulators
- On local PC (offline simulation)
- In/through the cloud (servers)

Execution in/through the cloud:





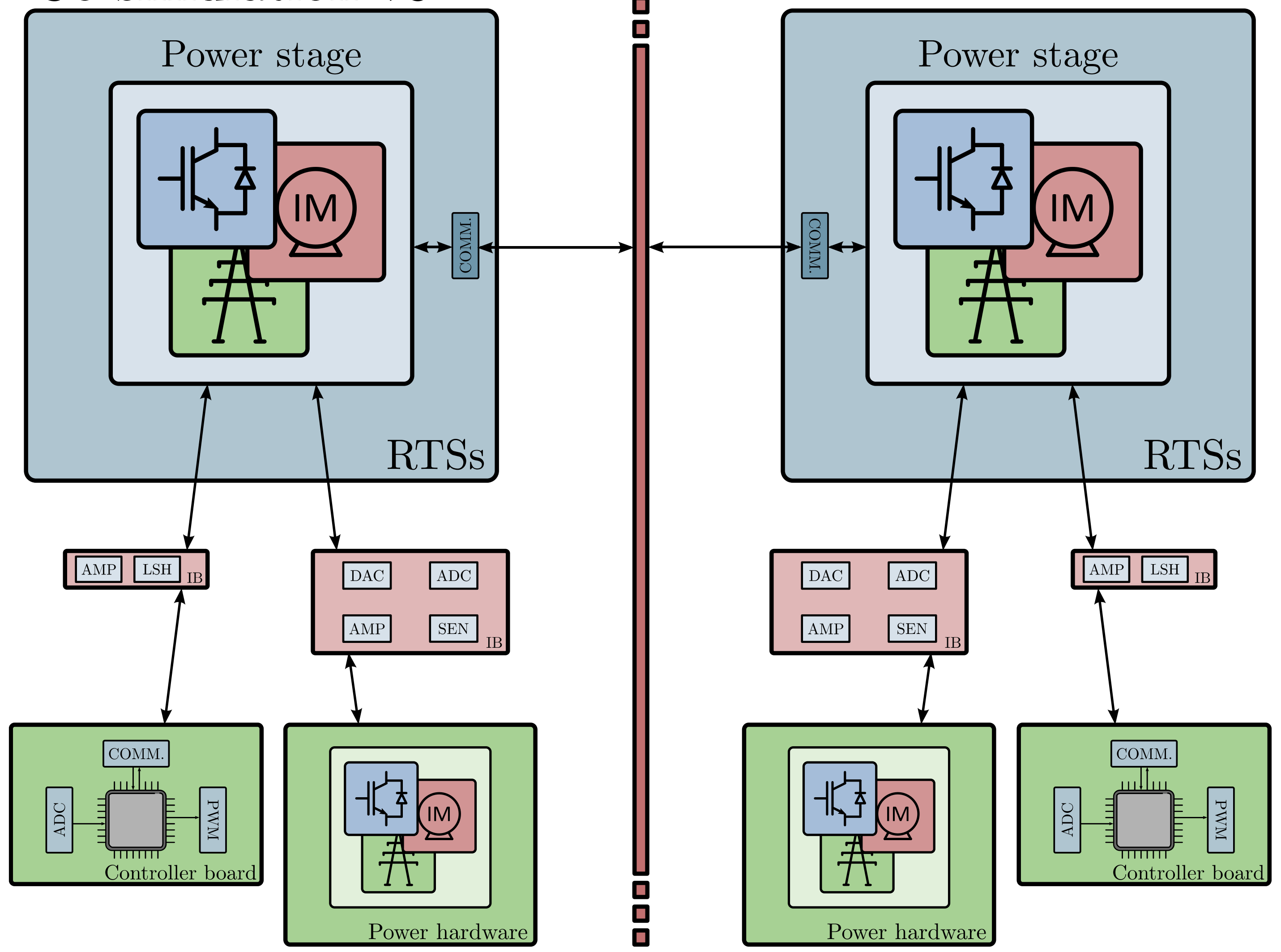
Results are coming in, but if emulators are used behind the cloud, the execution speed will be the same.

If the servers are used, even faster execution is expected (in comparison to the local PC usage).

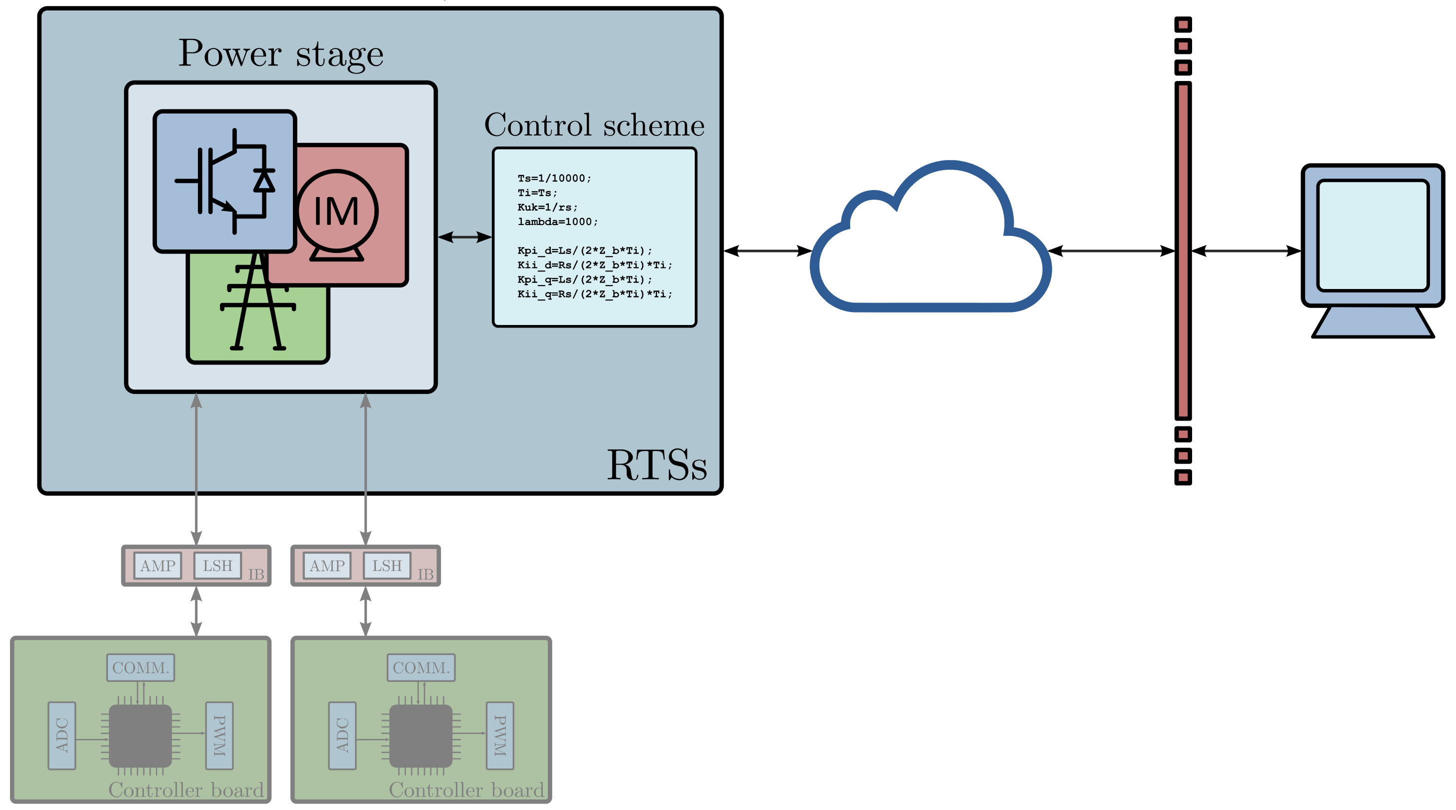
Conclusions

- Digital twinning has a relatively long and rich history
- Different twinning approaches are possible
- We are capable of emulating large and complex systems in real-time and even faster than real-time
- Integration with many cloud, and generally third-party, services now possible

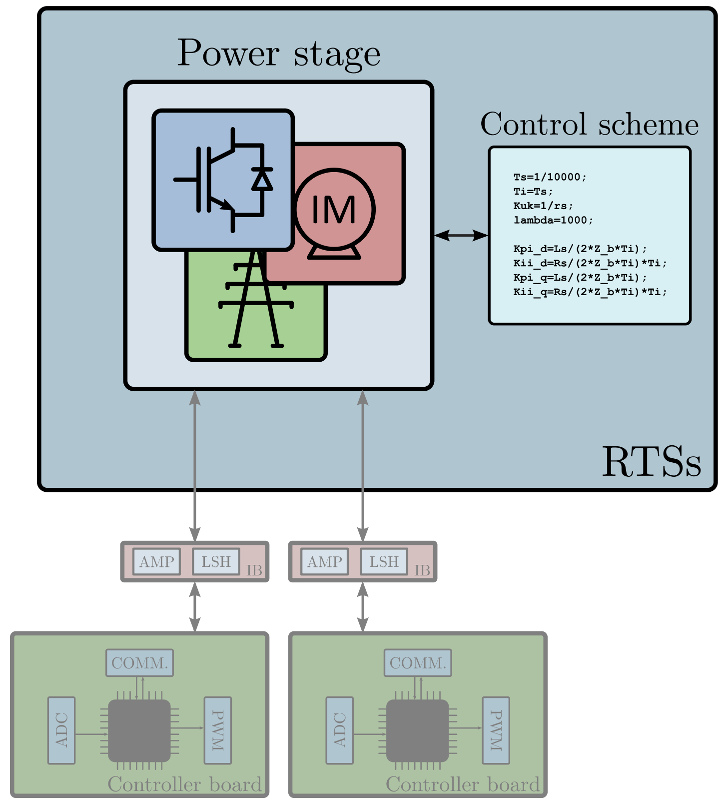
Co-simulation v3



Cloud emulation/simulation



Co-simulation v2.2



Thank you!

www.creates.energy

This project has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement n° 957815