How to choose your simulator, (and why SimGrid may be a good candidate) ?

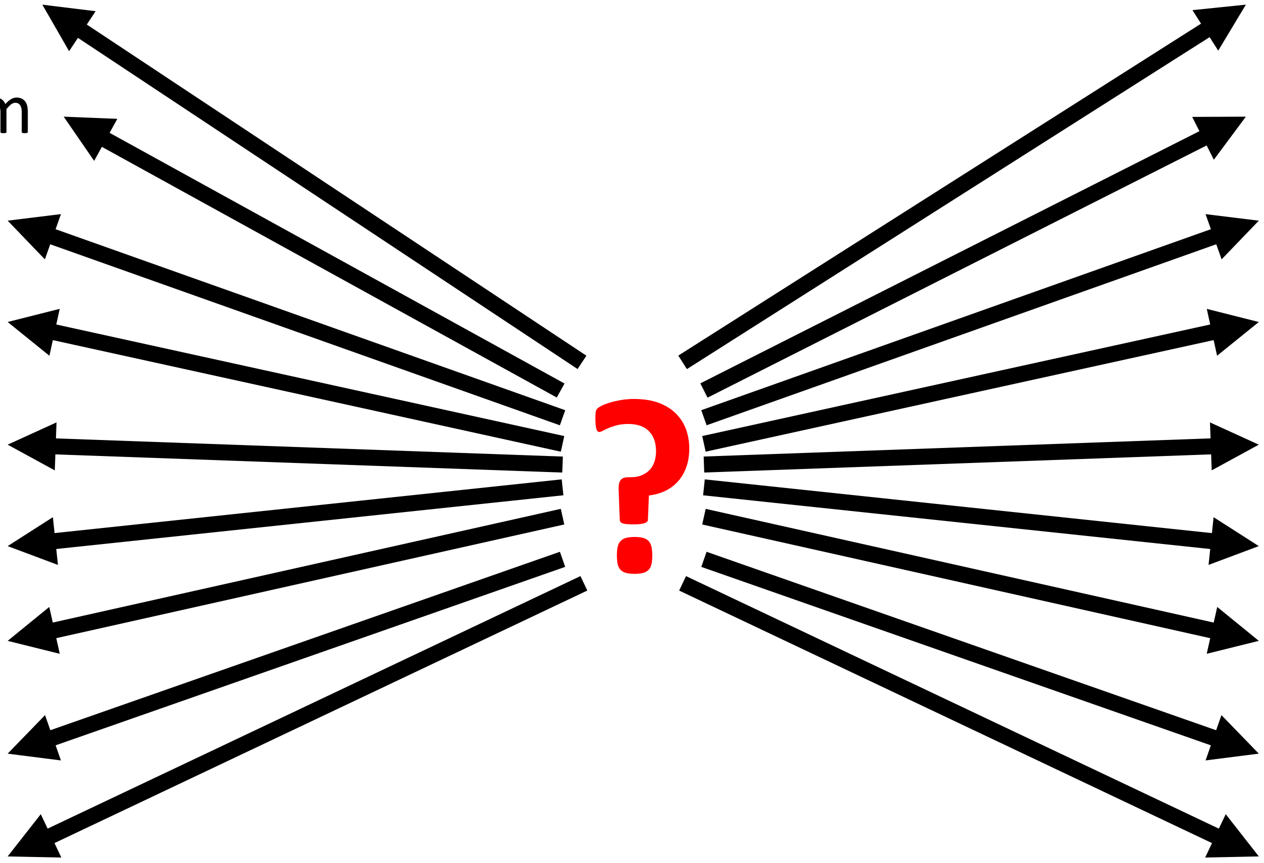
Benjamin CAMUS

Introduction

- Some conclusions of Martin's talk :
  - Building a simulator is a very complex task
  - Do NOT build your own simulator from scratch if an existing tools already fit your needs !

The jungle of the simulation tools

- PSINS
- LogGOPSim
- BigSim
- MPI-SIM
- OptorSim
- GridSim
- GroudSim
- CloudSim
- iCanCloud



- SimBA
- EmBOINC
- SimBOINC
- PeerSim
- OverSim
- DCSim
- SimGrid
- …
- Ad-hoc simulators…

Introduction

- Some conclusions of Martin’s talk :
  - Building a simulator is a very complex task
  - Do NOT build your own simulator from scratch if an existing tools already fit your needs !

Introduction

- Some conclusions of Martin’s talk :
  - Building a simulator is a very complex task
  - Do NOT build your own simulator from scratch if an existing tools already fit your needs !
- **Remaining questions :**
  - **How to choose your simulator ?**
  - **Why SimGrid may be a good candidate ?**

My background

- I'm using SimGrid in those projects:
  - **COSMIC** (2016 – 2018)
    - Goal: reduce the carbon footprint of distributed Cloud
    - Simulation of distributed Cloud + local PV power production
  - **RennesGrid** (2018 – 2019)
    - Goals:
      - Evaluate the impact of ITC infrastructure on SmartGrid
      - Propose an energy-efficient ITC infrastructure for RennesGrid
    - Simulation of a SmartGrid and its ITC infrastructure
  - **Energy-efficiency of workflows on Cloud** (2018)
    - Goals: investigate the impact of VM clustering on the energy-efficiency of workflow
    - Simulation of workflow executions on a data-center
- Thesis in Theory of Modeling and Simulation

How to choose your simulator ?

1. Does the modelling paradigm of the tool fit my needs ?
   - Design choices
2. Can I trust my simulation results ?
   - Validation & Verification
3. Can I connect the simulator with other tools ?
   - Simulator interoperability

Does the modelling paradigm fit my needs ?

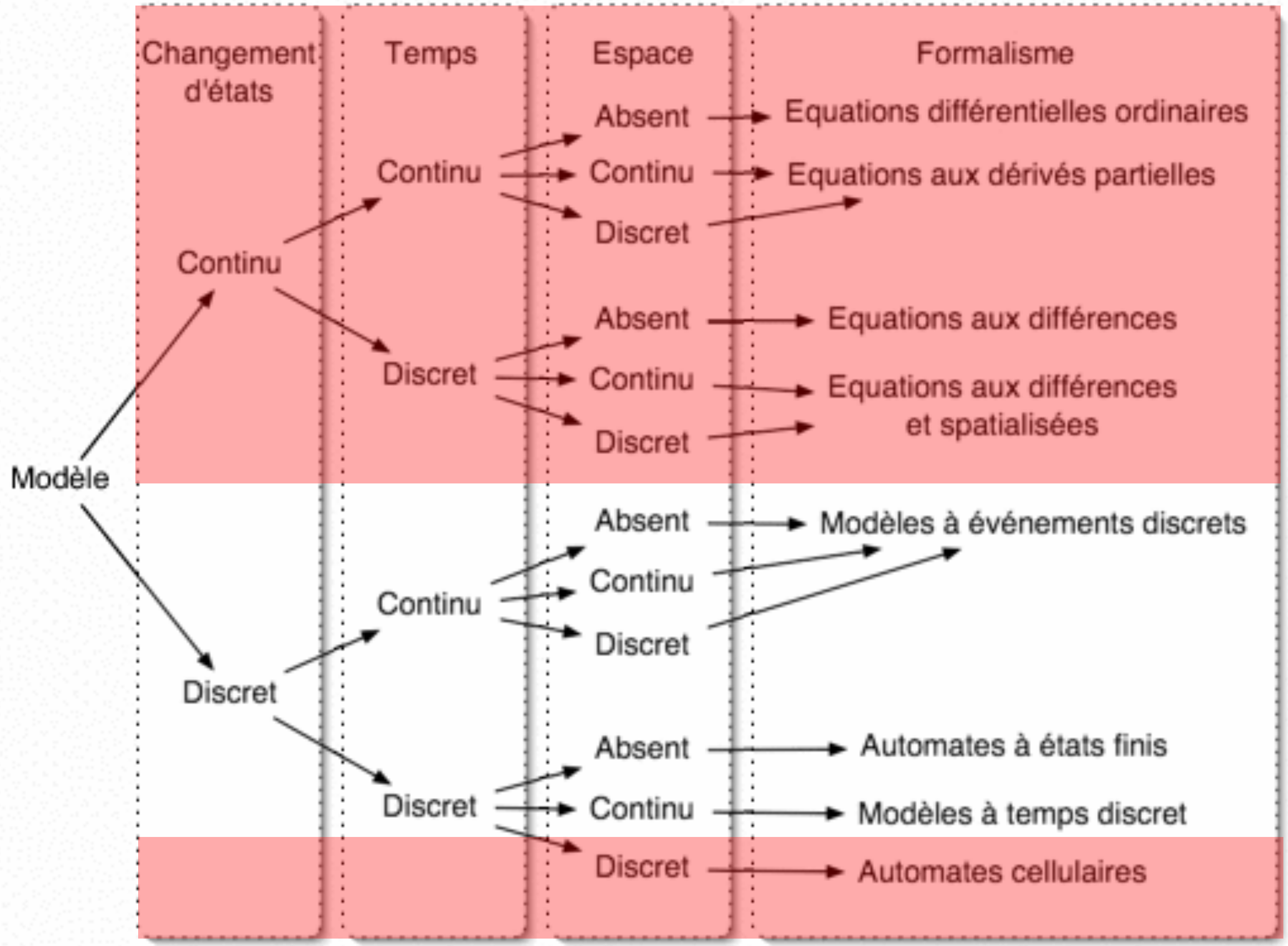
Design choices

Selection criteria

- Built-in models
  - Can I actually simulate what I want with the tool ?
- Accuracy
  - Can I have sufficiently accurate results?
- Modularity
  - Can I easily modify my systems ?
- Execution time
  - Can I run as many simulations as needed in a reasonable time ?

➔ **What do you want to do ?**
  - What do you want to study ?
  - What is your **experimental plan** ?

Modeling paradigms: execution policies



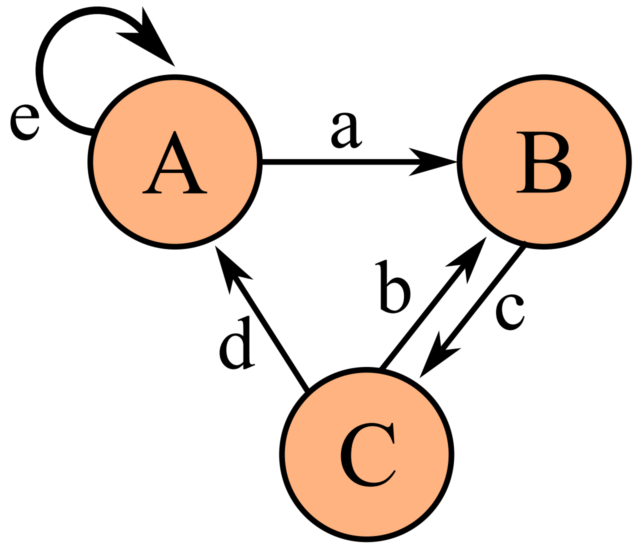
1

- Time-Stepped:
  - ✓ **Simple to design**
  - ✗ Approximation errors
  - ✗ Not efficient
- Event-based:
  - ✗ Complex design (error prone)
  - ✓ **Accurate**
  - ✓ **Efficient**

1. Ramat, E. (2006). Introduction à la modélisation et à la simulation à événements discrets. Modélisation et simulation multiagents : applications aux Sciences de l'Homme et de la Société, Lavoisier

Modeling paradigms: languages

- Formalism-based
  - ✓ **Modular**
  - ✓ **Simulation + Formal proof**
  - ✘ Rigid
  - ✘ Approximation errors
  - ✘ Only prototypes of complex applications

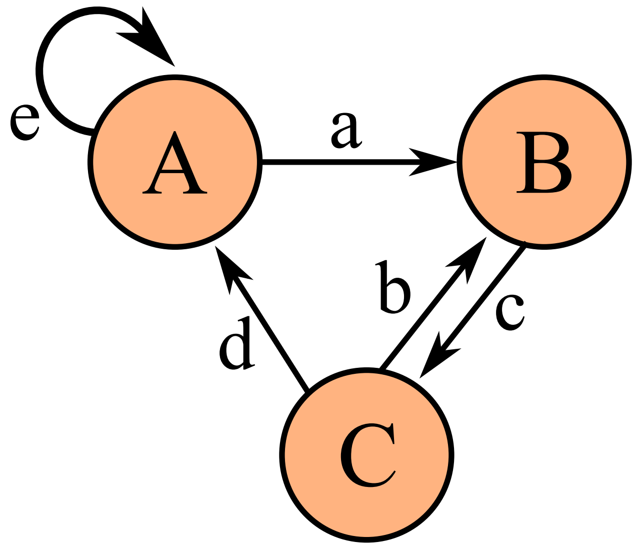


- Code-based
  - ✘ Only simulation
  - ✘ Less modular
  - ✓ **Flexible**
  - ✓ **More accurate**
  - ✓ **Test real application**

```
// worker-begin
static void worker(std::vector<std::string> args)
{
  xbt_assert(args.size() == 1, "The worker expects no argument");

  simgrid::s4u::Host* my_host      = simgrid::s4u::this_actor::get_host();
  simgrid::s4u::MailboxPtr mailbox = simgrid::s4u::Mailbox::by_name(my_host->get_name());

  double compute_cost;
  do {
    double* msg  = static_cast<double*>(mailbox->get());
    compute_cost = *msg;
    delete msg;

    if (compute_cost > 0) /* If compute_cost is valid, execute a computation of that cost */
      simgrid::s4u::this_actor::execute(compute_cost);

  } while (compute_cost > 0); /* Stop when receiving an invalid compute_cost */

  XBT_INFO("Exiting now.");
}
```

Modeling paradigms: languages

- Formalism-based
  - ✓ **Modular**
  - ✓ **Simulation + Formal proof**
  - ✗ Rigid
  - ✗ Approximation errors
  - ✗ Only prototypes of complex applications



- Code-based
  - ✗ Only simulation
  - ✗ Less modular
  - ✓ **Flexible**
  - ✓ **More accurate**
  - ✓ **Test real application**

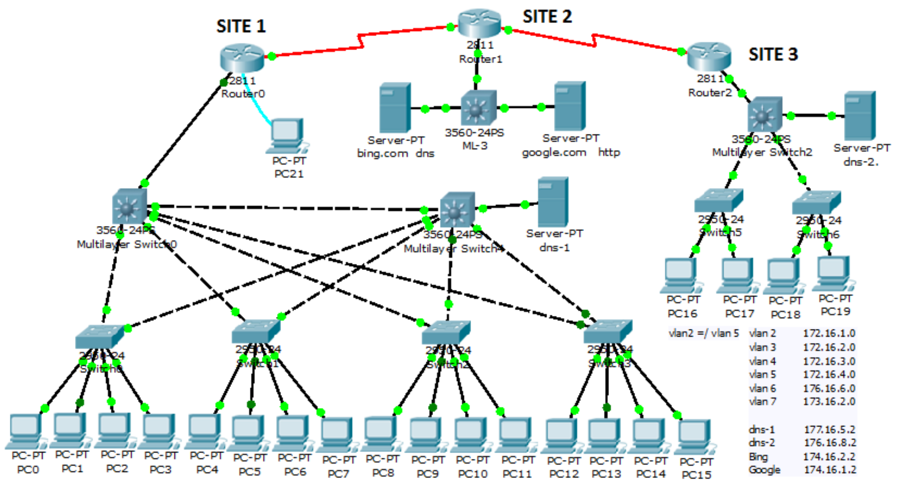
SIMGRID

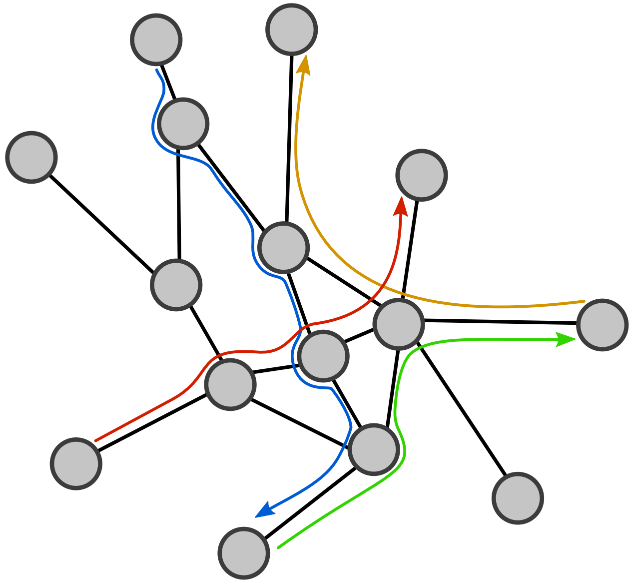
```
// worker-begin
static void worker(std::vector<std::string> args)
{
  xbt_assert(args.size() == 1, "The worker expects no argument");

  simgrid::s4u::Host* my_host      = simgrid::s4u::this_actor::get_host();
  simgrid::s4u::MailboxPtr mailbox = simgrid::s4u::Mailbox::by_name(my_host->get_name());

  double compute_cost;
  do {
    double* msg  = static_cast<double*>(mailbox->get());
    compute_cost = *msg;
    delete msg;

    if (compute_cost > 0) /* If compute_cost is valid, execute a computation of that cost */
      simgrid::s4u::this_actor::execute(compute_cost);

  } while (compute_cost > 0); /* Stop when receiving an invalid compute_cost */

  XBT_INFO("Exiting now.");
}
```

Modeling paradigms: representation level

- Fine-grain description
  - ✓ **More accurate simulation results**
  - ✓ **Answer to a larger set of questions**
  - ✘ Complex (error prone) simulation set-up
  - ✘ (very) slow simulations
- Coarse-grain description
  - ✘ Approximation
  - ✘ Answer to a smaller set of questions
  - ✓ **Simple simulation set-up**
  - ✓ **Faster simulations**

- Representation level = critical for
  - simulation performance
  - simulation accuracy

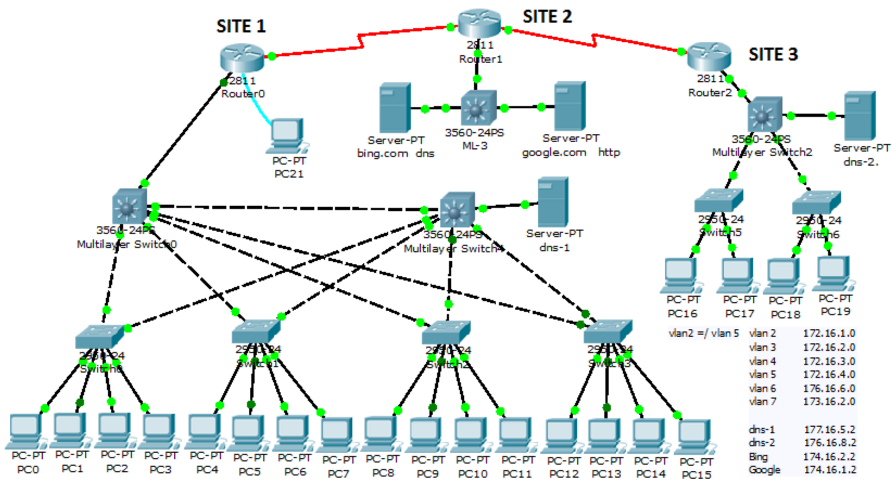


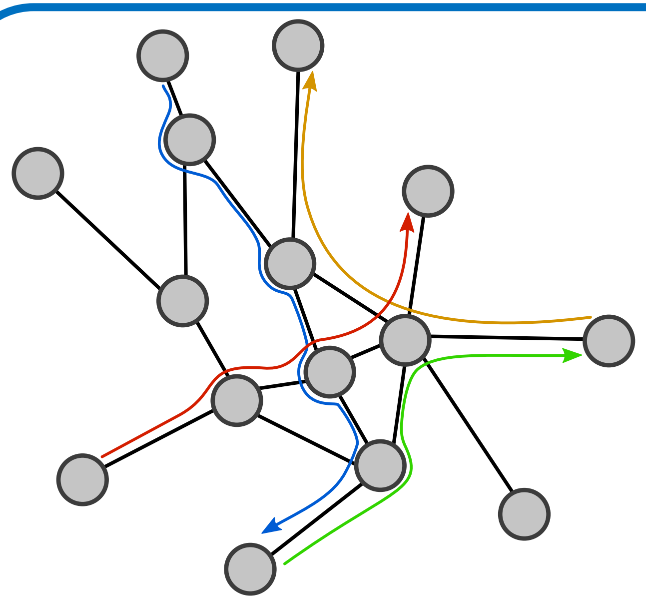


Courtesy of Loic Guegan

Modeling paradigms: representation level

- Fine-grain description
  - ✓ More accurate simulation results
  - ✓ Answer to a larger set of questions
  - ✗ Complex (error prone) simulation set-up
  - ✗ (very) slow simulations
- Coarse-grain description
  - ✗ Approximation
  - ✗ Answer to a smaller set of questions
  - ✓ Simple simulation set-up
  - ✓ Faster simulations





Courtesy of Loic Guegan

- Representation level = critical for
  - simulation performance
  - simulation accuracy

Can I trust my simulation results?

Validation & Verification

Models validation

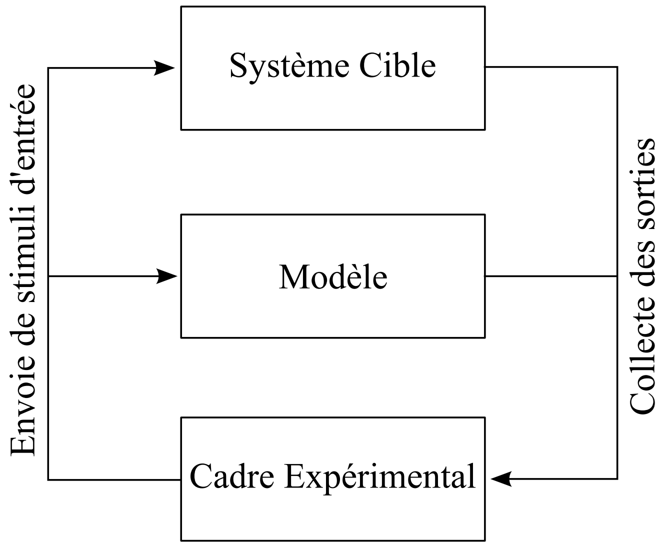
- *« A **model** is a simplification of a system under study »*[1]
- Simplify to :
  - Make experiment feasible
  - Decrease experimentations costs
  - Ease experimentation set-up
  - Speed up experiments
- *« Model **validation** is substantiating that the model, within its domain of applicability, behaves with satisfactory accuracy consistent with the M&S objectives »*[1]
  - i.e. Is my simplification sufficiently correct ?

---

1. Balci, O. (1998). Verification, validation, and accreditation. In Proceedings of the 30th conference on Winter simulation, pages 41–4. IEEE Computer Society Press.

Validation in practice

- In practice, different understandings of validation :
  - ✘ Models dynamics ≈ theoretical dynamics
    - Theory ≠ reality
  - ✘ Simulation results ≈ expected results
    - Expectation ≠ reality
  - ✓ **Models results ≈ real system measurements**



- Validation is only relative to an **experimental frame**[1]
  - Define the condition of observation of the system
  - Bound the validity of the model
  - Often only implicitly defined

1. Zeigler, B., Praehofer, H., and Kim, T. (2000). Theory of Modeling and Simulation : Integrating Discrete Event and Continuous Complex Dynamic Systems. Academic Press.

Validation of SimGrid models

- Validated models for:
  - CPU sharing
    - *P. Velho, "Accurate and Fast Simulations of Large-Scale Distributed Computing Systems," Theses, Université Grenoble Alpes, Jul. 2011.*
  - TCP
    - *P. Velho, L. Schnorr, H. Casanova, and A. Legrand, "On the Validity of Flow-level TCP Network Models for Grid and Cloud Simulations," ACM Trans. on Modeling and Computer Simulation, 2013.*
  - Energy
    - *F. C. Heinrich, T. Cornebize, A. Degomme, A. Legrand, A. Carpen-Amarie, S. Hunold, A.-C. Orgerie, and M. Quinson, "Predicting the Energy Consumption of MPI Applications at Scale Using a Single Node," in IEEE Cluster, Sep. 2017.*
  - Virtual machines
    - *L. Pouilloux, T. Hirofuchi, and A. Lebre, "SimGrid VM: Virtual Machine Support for a Simulation Framework of Distributed Systems," IEEE Trans. on Cloud Computing, 2015.*
- **Experimental frame:** some limits are clearly identified ➔ **Read the papers**

Models verification

- « ***Model Verification** is substantiating that the model is transformed [into an executable program] with sufficient accuracy. »*[1], e.g. :
  - Does my simulation method execute my model in an accurate way ?
  - Does my program have bugs ?

- **Validation vs. Verification** :
  - « *Model validation deals with building the right model* »[1]
  - « *Model verification deals with building the model right* »[1]

- Should be performed all along the software life-cycle

---

1. Balci, O. (1998). Verification, validation, and accreditation. In Proceedings of the 30th conference on Winter simulation, pages 41–4. IEEE Computer Society Press.

Verification in practice

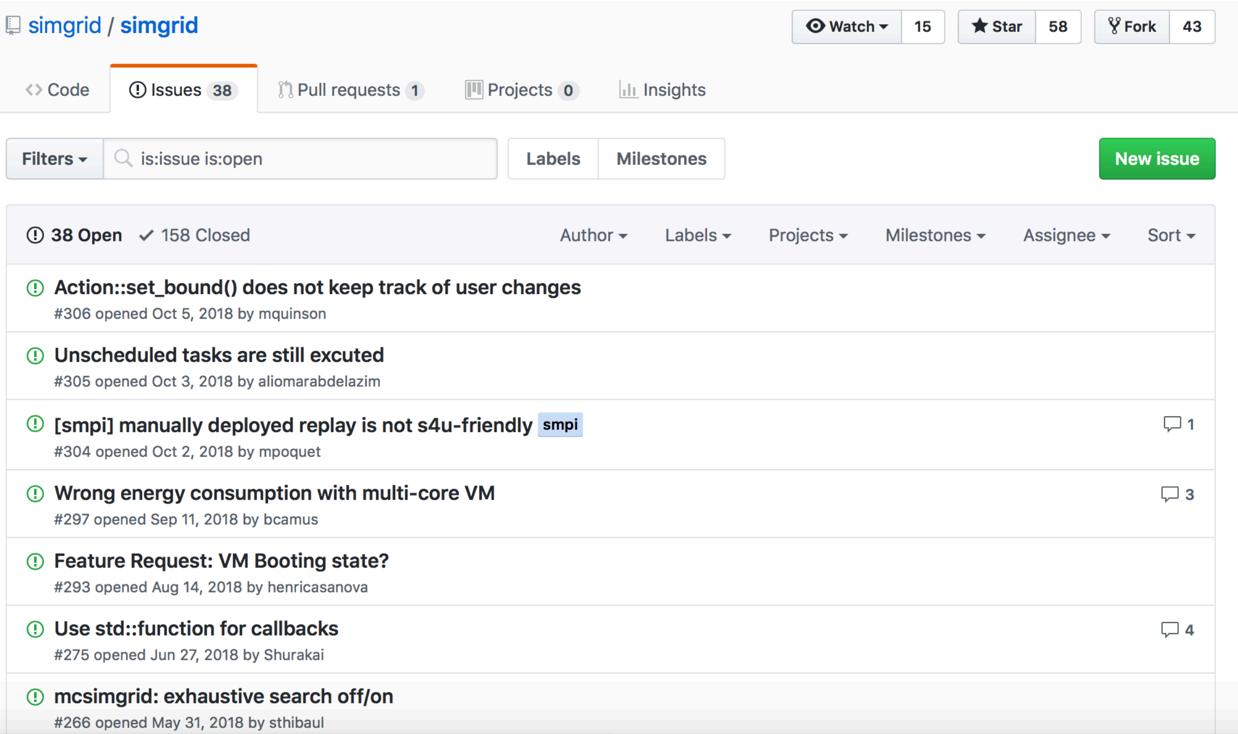
- Complete verification is rarely done, because
  - Simulation software are too complex
  - And in continuous evolution

- Is the code of the simulator verified ?
  - No guarentee 🡐 Most simulators
  - **Yes, here are the proofs 🡐 Few very simple simulators**
  - **Partially, here are the state of the verification process 🡐 reasonable answer**

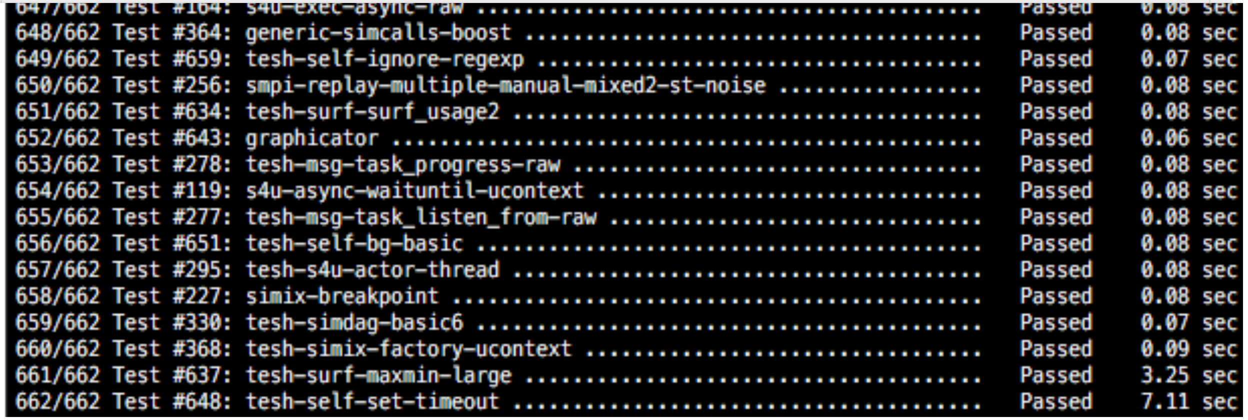
Verification with SimGrid

- Does SimGrid have bugs ?
  - Off course !
  - But they are:
    - (Partially) identified
    - Publicly accessible

➔ **Check if some bugs may have an impact on the validity of your simulation results**



- Continuous integration :
  - Keep a good level of verification,
  - which can only increase

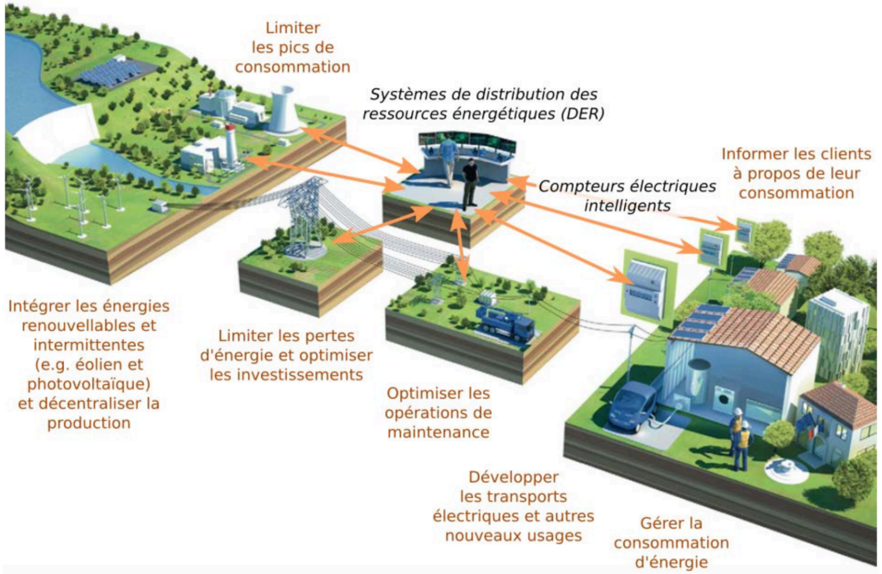


Can I connect the simulator with other tools?

Modelling tools interoperability

Co-simulation

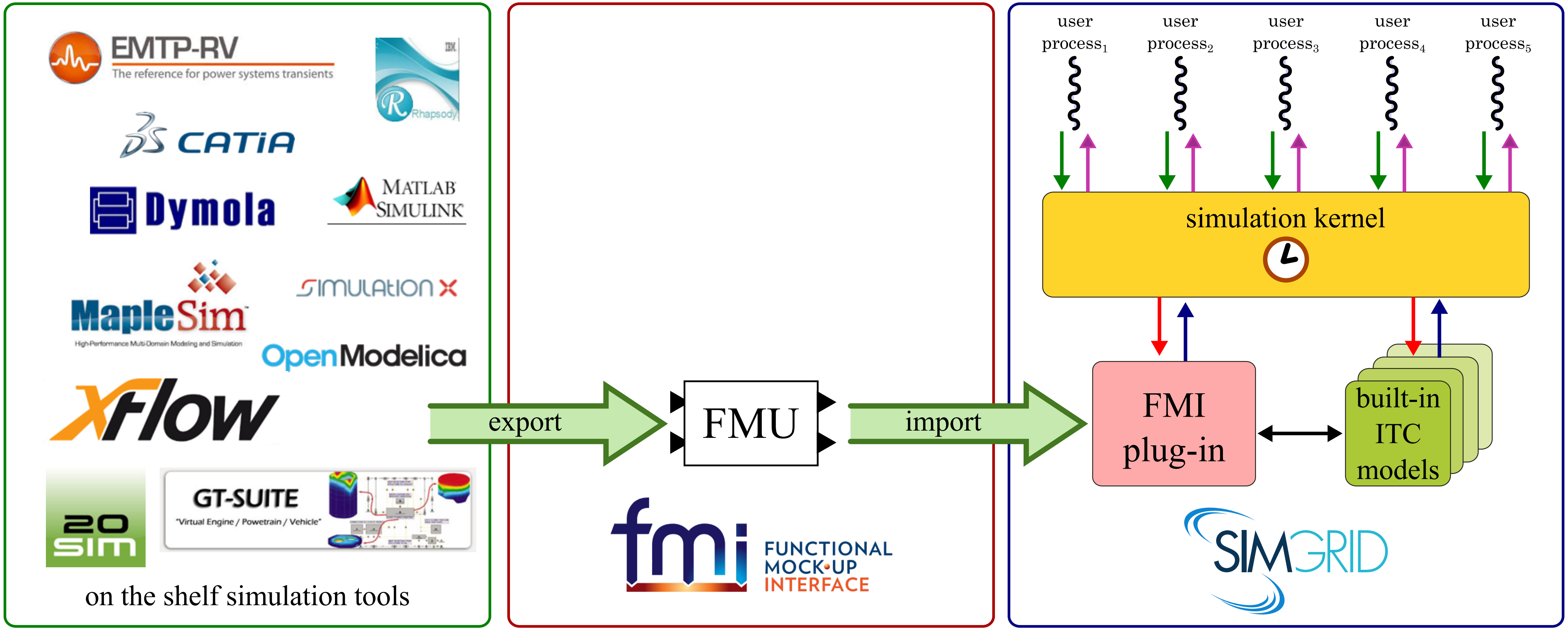
- Interact with other simulation tools
- To use the advantages of different distributed ITC simulators
- To simulate interactions between **distributed ITC** and **physical systems**, e.g:
  - data-center and its cooling system
  - Smart Grid and their ITC systems
  - IoT devices and their environments
  - …



- **Challenges:**
  - Software interoperabily
  - Integrate heterogeneous execution policies
  - Bridge the gap between code-based and formal-based models

Co-simulation with SimGrid

- SimGrid plug-in to support the FMI standard[1]



1. *B. Camus, A-C Orgerie, and M. Quinson. Co-simulation of FMUs and Distributed Applications with SimGrid. In SIGSIM-PADS '18, ACM(2018).*

Conclusion : How to choose your simulator ?

- **Not trivial**
- Does the modelling paradigm fit my needs ?
  - Depends on:
    - What do you want to **study** on the system
    - Your **experimental plan**
  - Check:
    - Modelling paradigm
    - Execution policy
    - Representation level
- Can I trust my simulation results ? = check for:
  - Model **validity** (and **experimental frame**)
  - Model **verification**
- Check for simulator **interoperability**