Université Nice Sophia Antipolis

i3S sophia antipolis

Inria
INVENTEURS DU MONDE NUMÉRIQUE

TimeSquare

a tool based on logical time for the modeling of real time embedded systems

Julien DeAntoni (some of the slide's are taken from Frédéric Mallet's one)

EQUIPE PROJET
AOSTE
CENTRE Sophia Antipolis Méditerranée

Université Nice Sophia Antipolis

i3S sophia antipolis

Inria
INVENTEURS DU MONDE NUMÉRIQUE

TimeSquare

a tool based on logical time for the modeling of real time embedded systems

Julien DeAntoni (some of the slide's are taken from Frédéric Mallet's one)

EQUIPE PROJET
AOSTE
CENTRE Sophia Antipolis Méditerranée

AGENDA

1. Logical Time
2. CCSL
3. TimeSquare
4. Démonstration

1

Logical Time

Logical Time...

- focuses on causal relations between events
- Is independent of the abstraction level
- Is multi-clock (polychronous)
- Any event can be taken as a reference
- Provides a partial order between events

After 23 starts of the computer, a disk check is done

The computation duration is 156 processor ticks

Logical Time...

- focuses on causal relations between events
- Is independent of the abstraction level
- Is multi-clock (polychronous)
- Any event can be taken as a reference
- Provides a partial order between events

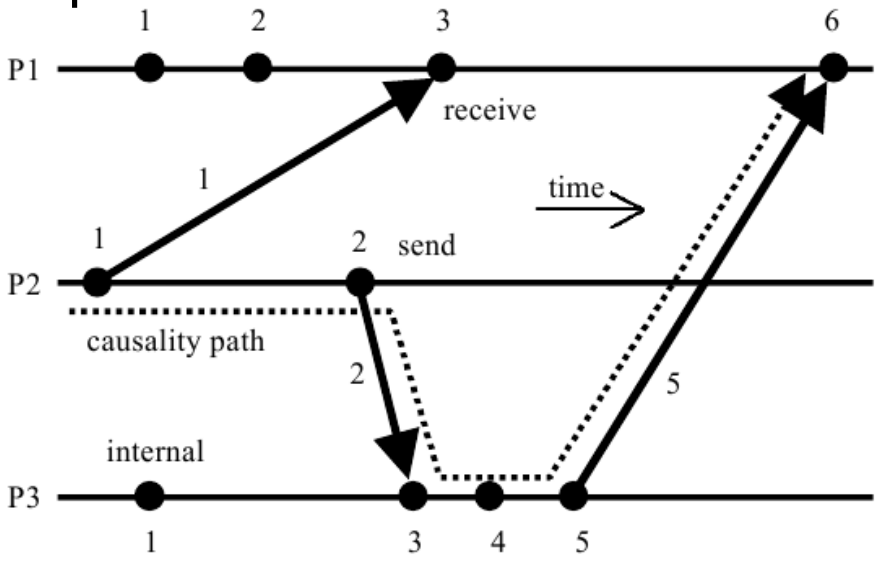
After 23 starts of the computer, a disk check is done

The computation duration is 156 processor ticks

In modern laptop, tick is logical since the processor speed depend on the battery level

Logical Time...

- focuses on causal relations between events
- Is independent of the abstraction level
- Is multi-clock (polychronous)
- Any event can be taken as a reference
- Provides a partial order between events



LOGICAL TIME
Friedemann Mattern
Darmstadt University of Technology

Logical Time and Physical Time

Logical functional time

- Functional, from the early design steps

Action1 causes by Action2

A check disk is done every 23 starts

“Physical” time

- Extra functional, appears lately in the development process

Action1 must be done in less than 30ms

A disk check mus be done in less than 45 seconds

Logical Time and Physical Time

Logical functional time

- Functional, from the early design steps
- Multiple times (local / global)

Neither relative activation rates nor regularity are forced, YET

Task1 is periodic every 300 ECU1 ticks
Task2 is periodic every 200 ECU2 ticks

"Physical" time

- Extra functional, appears lately in the development process
- Single time

Relative activation rates and regularity are forced

Task1 is periodic every 10 ms
Task2 is periodic every 30ms

Logical Time and Physical Time

Logical functional time

- Functional, from the early design steps
- Multiple times (local / global)
- Flexible

"Physical" time

- Extra functional, appears lately in the development process
- Single time
- Rigid

Most of the time only relative rate matters,
Exact timing may be given latter, when many parameters are known

Task2 is periodic every 3 Task1

Task1 is periodic every 10 ms
Task2 is periodic every 30ms

Logical Time and Physical Time

Logical functional time

- Functional, from the early design steps
- Multiple times (local / global)
- Flexible
- Able to specify physical time

After 5 events of the "second" clock, it stops

"Physical" time

- Extra functional, appears lately in the development process
- Single time
- Rigid
- A special case of logical time

After 5 seconds, it stops...

Logical Time and Physical Time

Logical functional time

- Functional, from the early design steps
- Multiple times (local / global)
- Flexible
- Able to specify physical

Systematic propagation of rates...

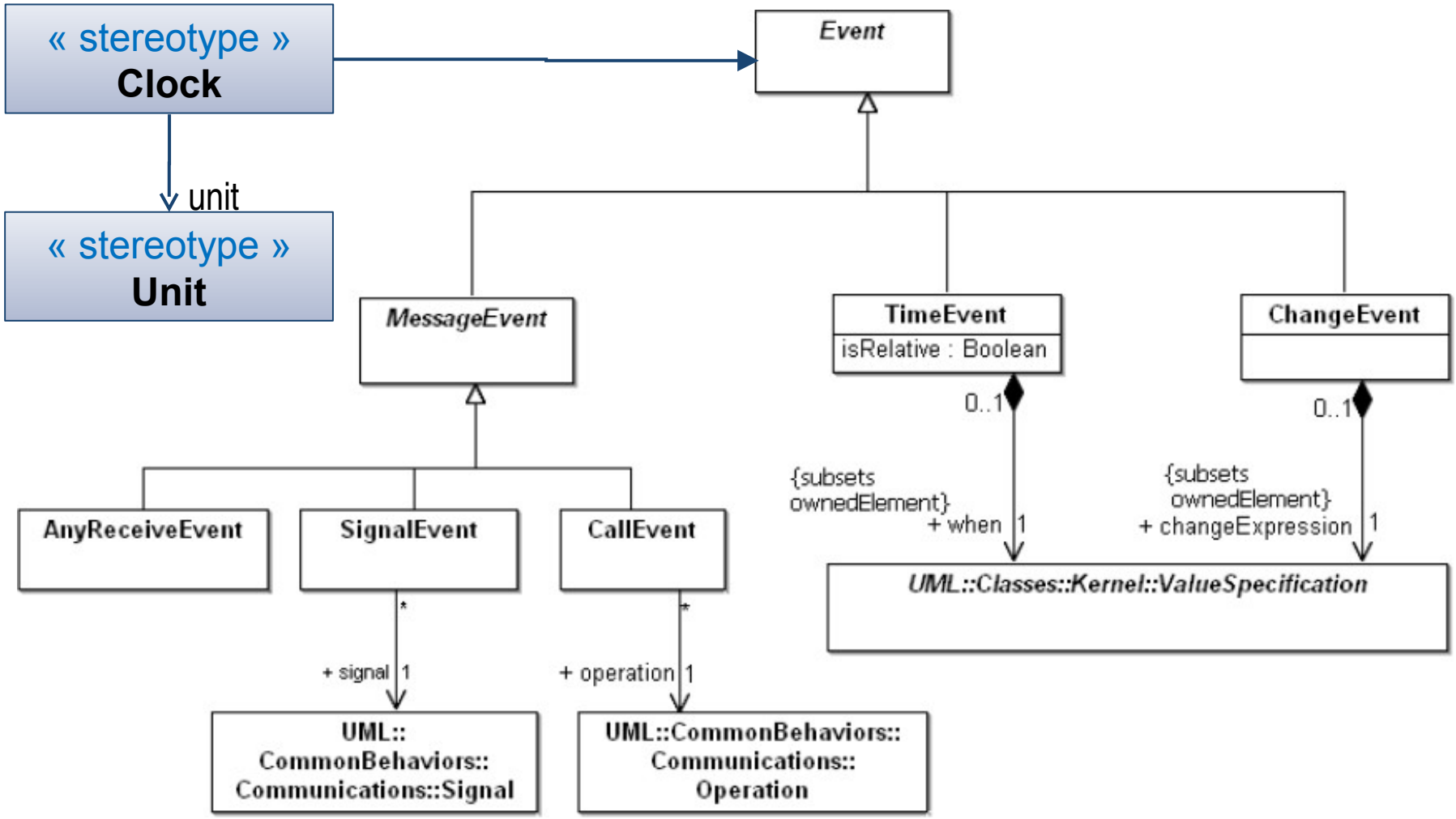
Task2 is periodic every 3 Task1
Task1 is periodic every 10 ticks of the "ms" clock

"Physical" time

- Extra functional, appears lately in the development process
- Single time
- Rigid
- A special case of logical time

Task1 is periodic every 10 ms
Task2 is periodic every 30ms

MARTE Time model is made for logical time



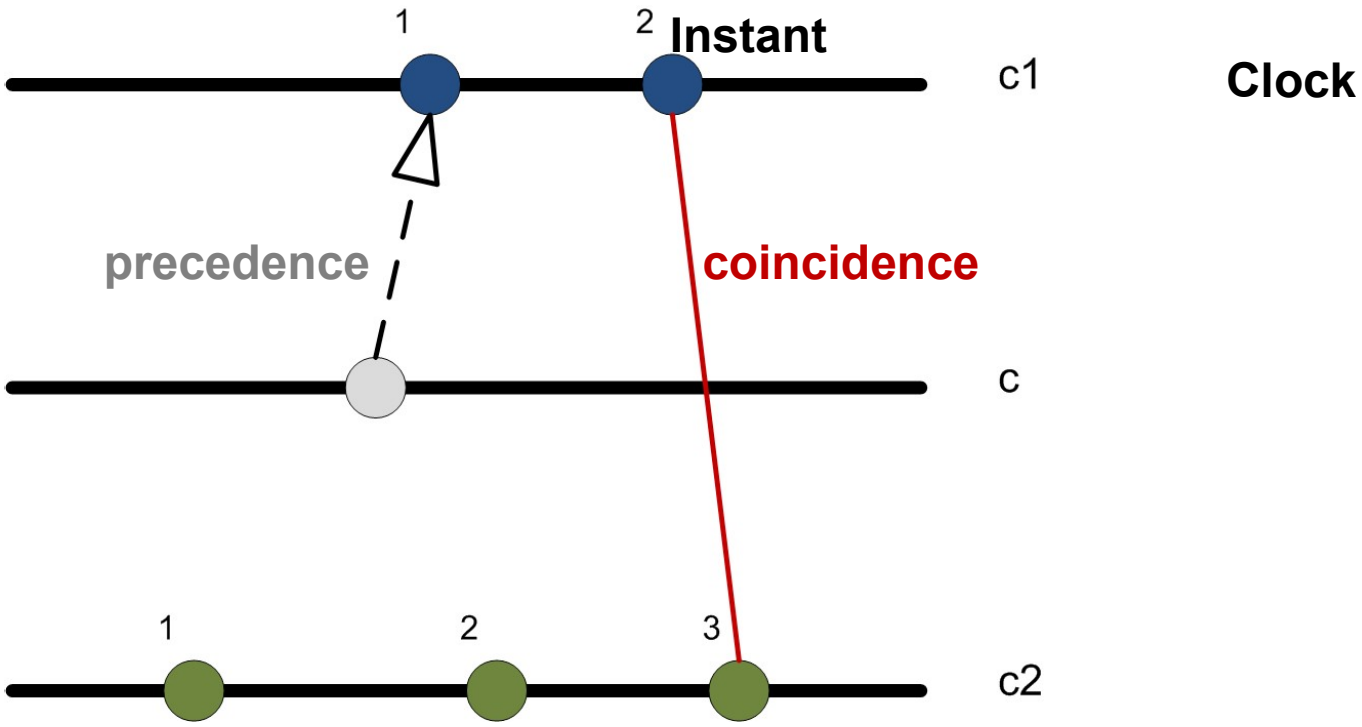
2

CCSL

Clock Constraint Specification Language

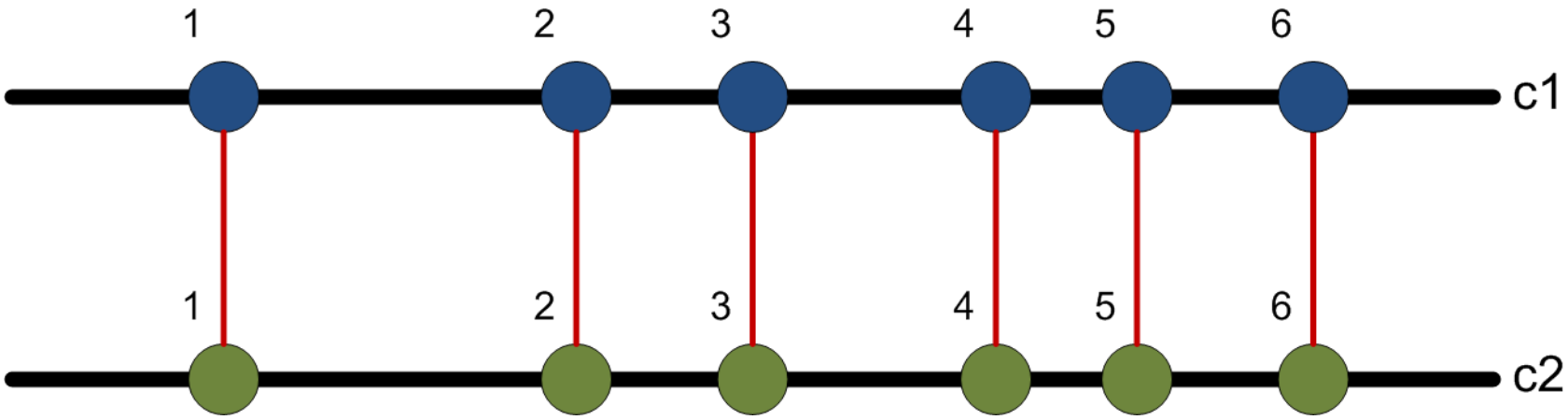
Logical clocks – instant relations

Clocks are *a priori* independent



Clock relations – Coincidence-based

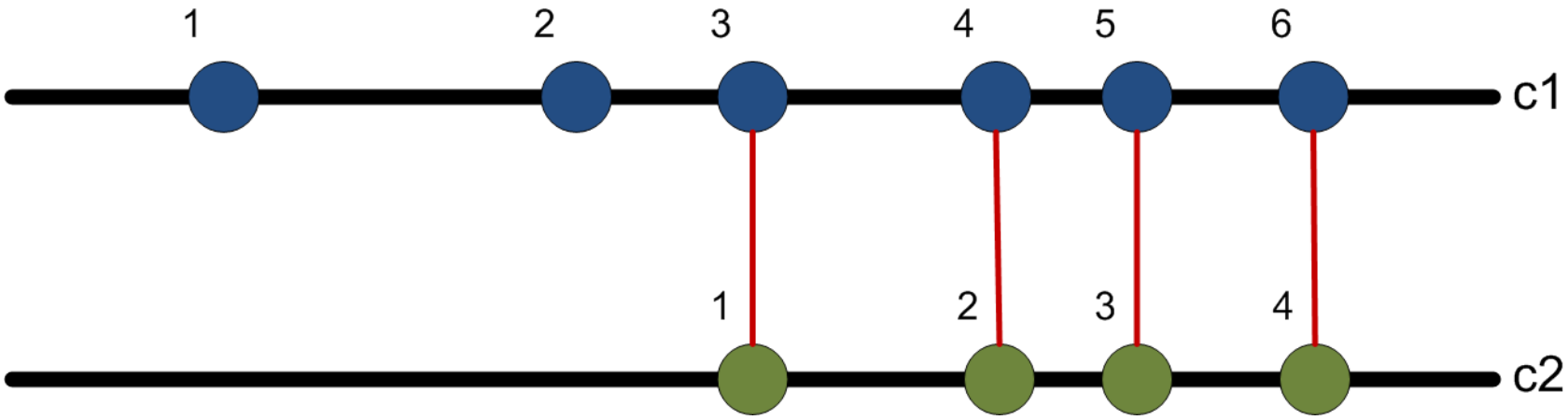
Infinitely many coincidence relations



c2 ⊑ c1

Clock relations – Coincidence-based

Infinitely many coincidence relations

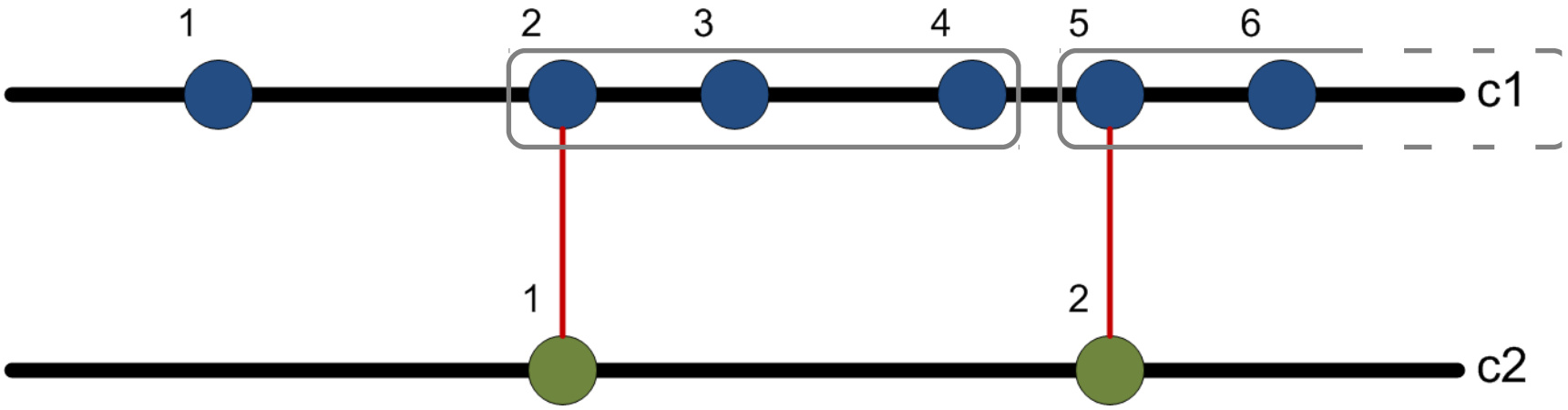


c2 ⊑ c1 $ 2

c2 is a **subclock** of c1

Clock relations – Coincidence-based

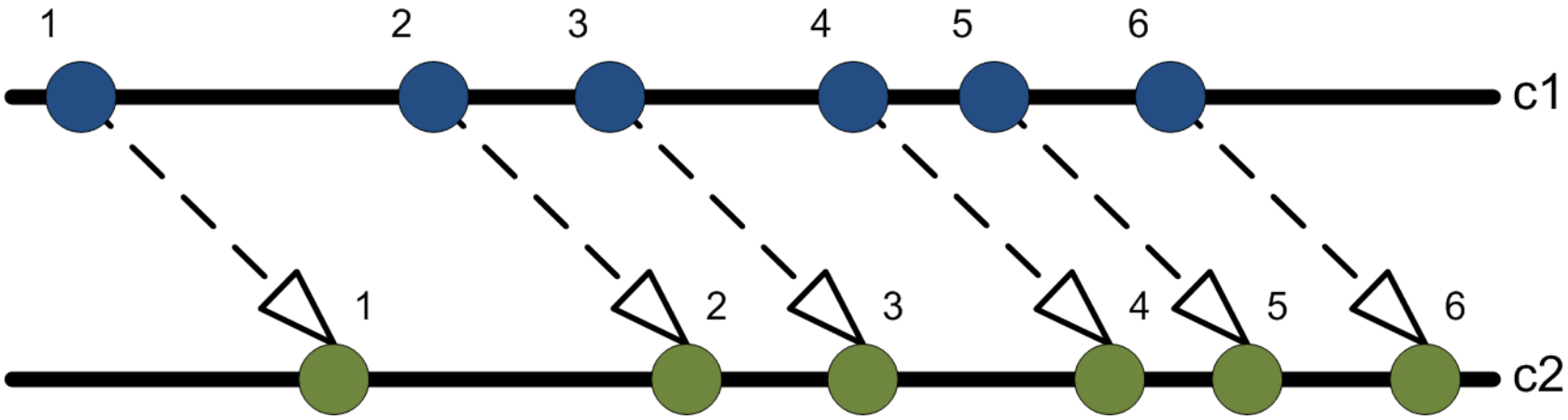
Infinitely many coincidence relations



c2 isPeriodicOn c1 period=3 offset=1

Clock relations – Precedence-based

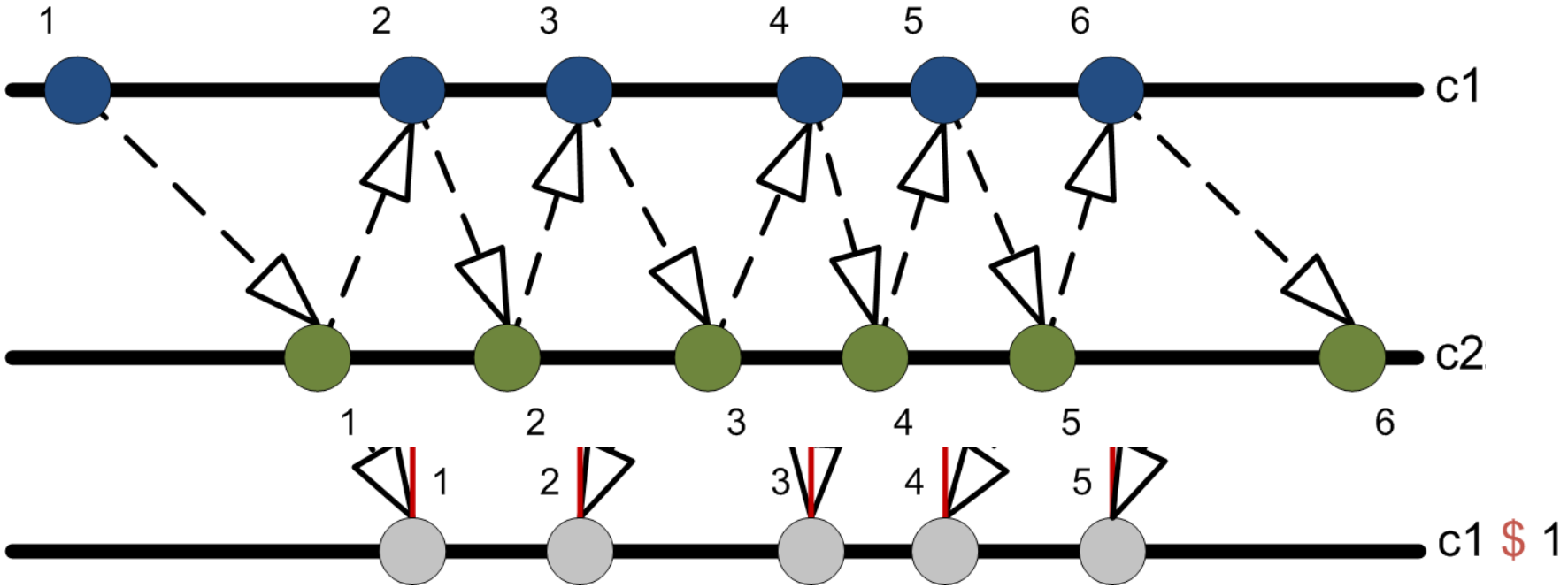
Infinitely many precedence relations



$c1 \leqslant c2$

Clock relations – Precedence-based

Can be bounded if required (for analysis)



c1 ⩽ c2 c1 $ 1 → c1 ~ c2

Kernel + Libraries

Clock relations - summary

Elementary relations

- Coincidence, precedence
- Mixed relations (sampling, delay)

Combined to build common time patterns

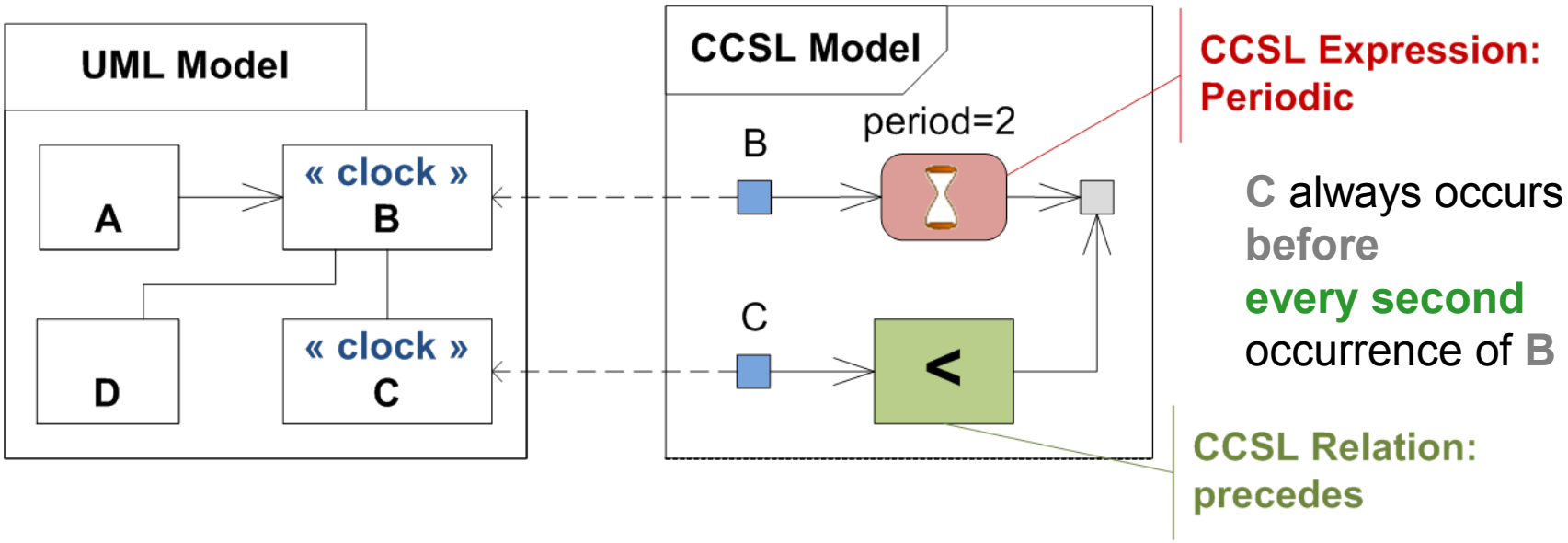
- Periodicity |a[i+1]-a[i]| = period
- Sporadicity |a[i+1]-a[i]| > interArrival
- Deadline |end[i]-start[i]| < deadline
- Jitter, skew, …

3

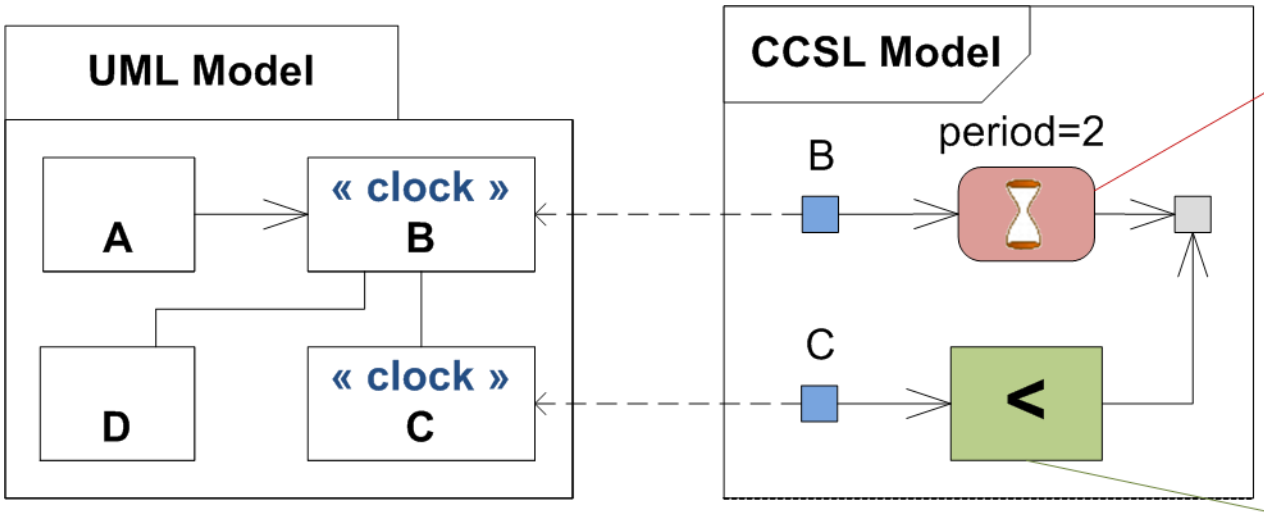
TimeSquare

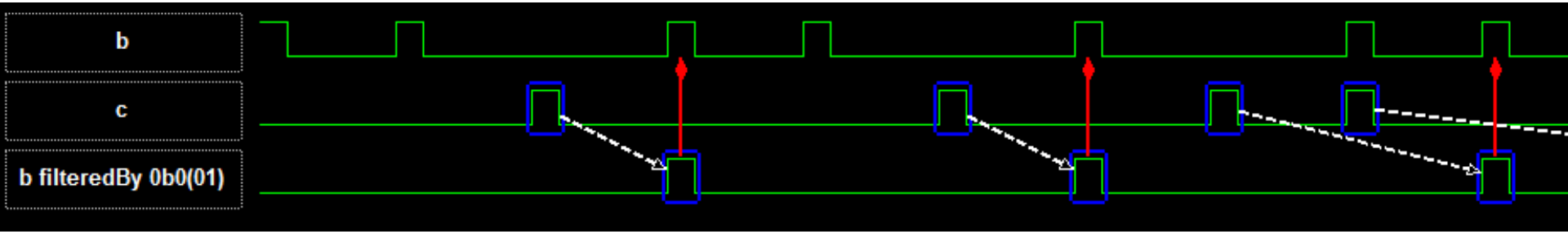
Analysis of MARTE/CCSL specifications... and more ;)

Annotate UML models
Identify and constrain clocks



Annotate UML models for simulation and animation

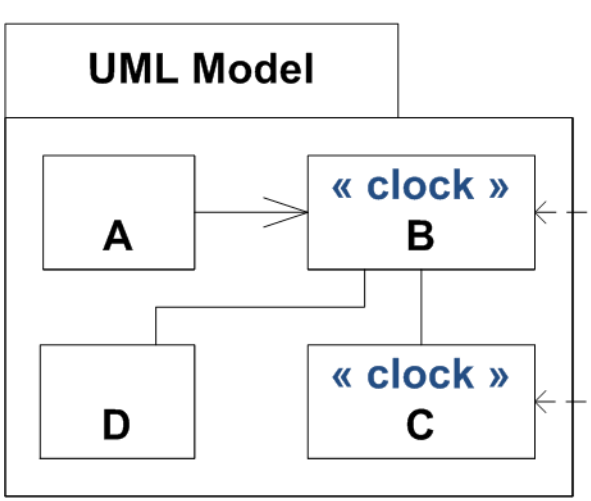


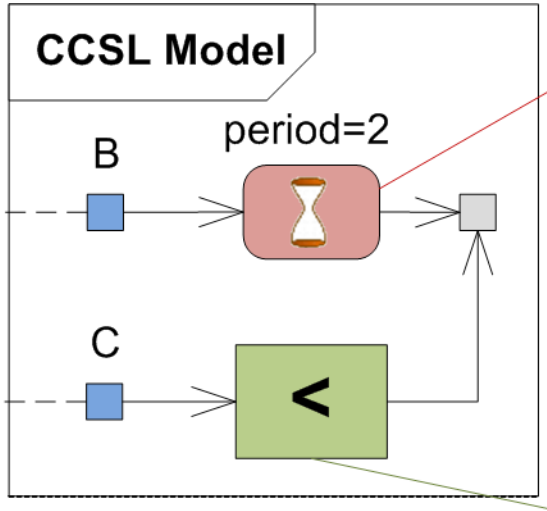


http://timesquare.inria.fr

TimeSquare

Adorns UML models for verifying requirements

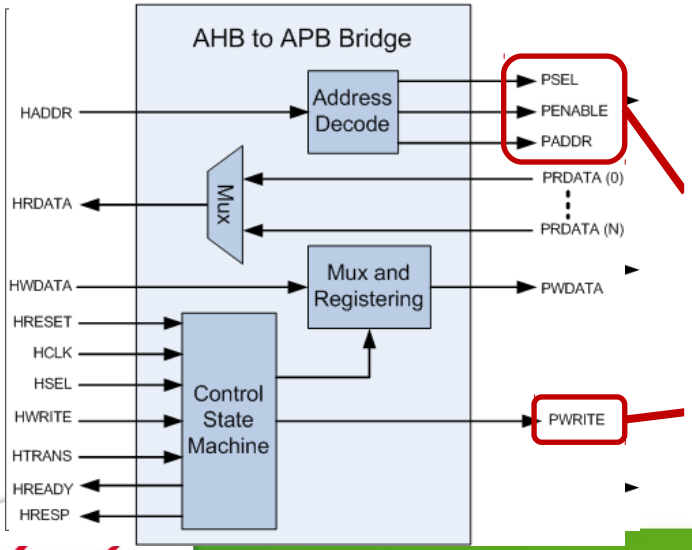


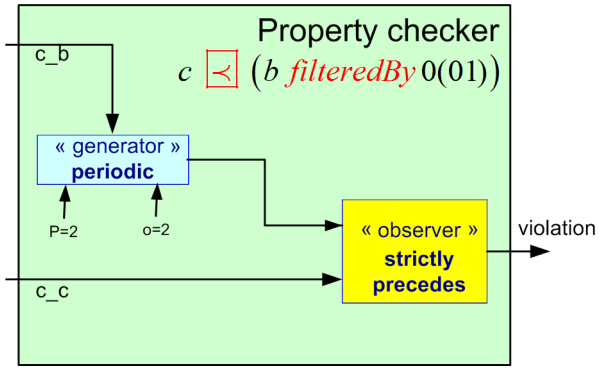


CCSL Expression: Periodic

specification

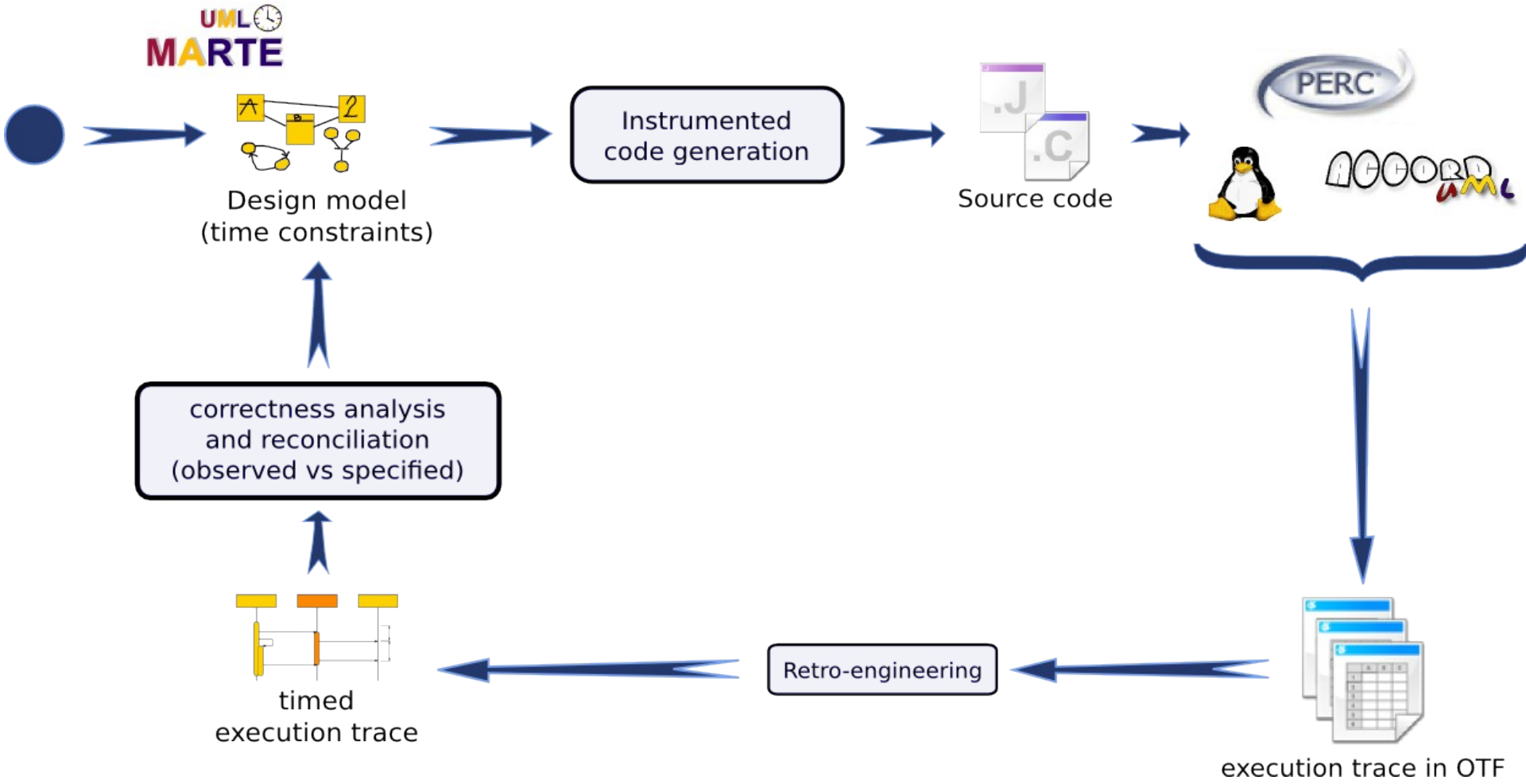
CCSL Relation: precedes



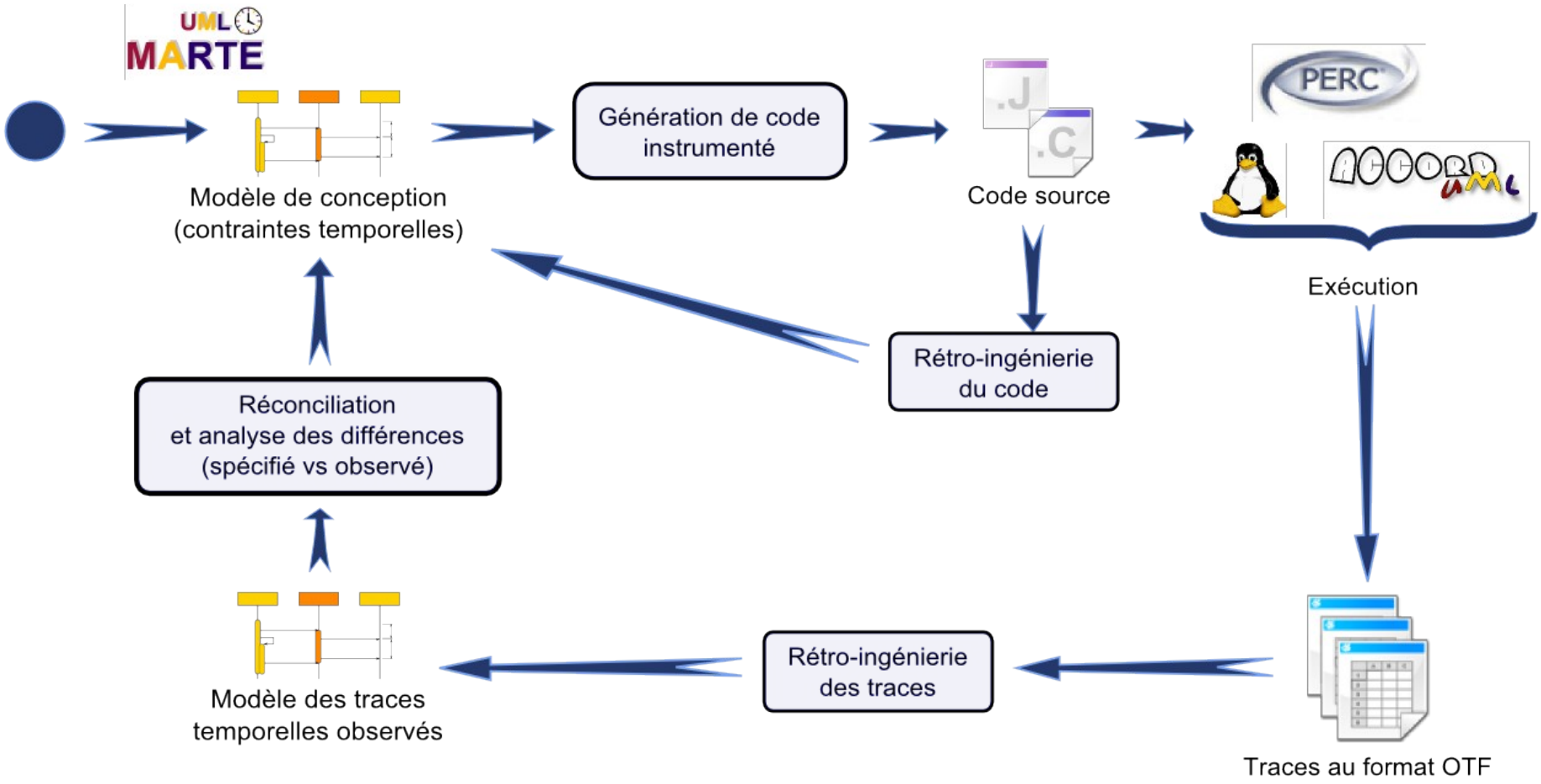


observation

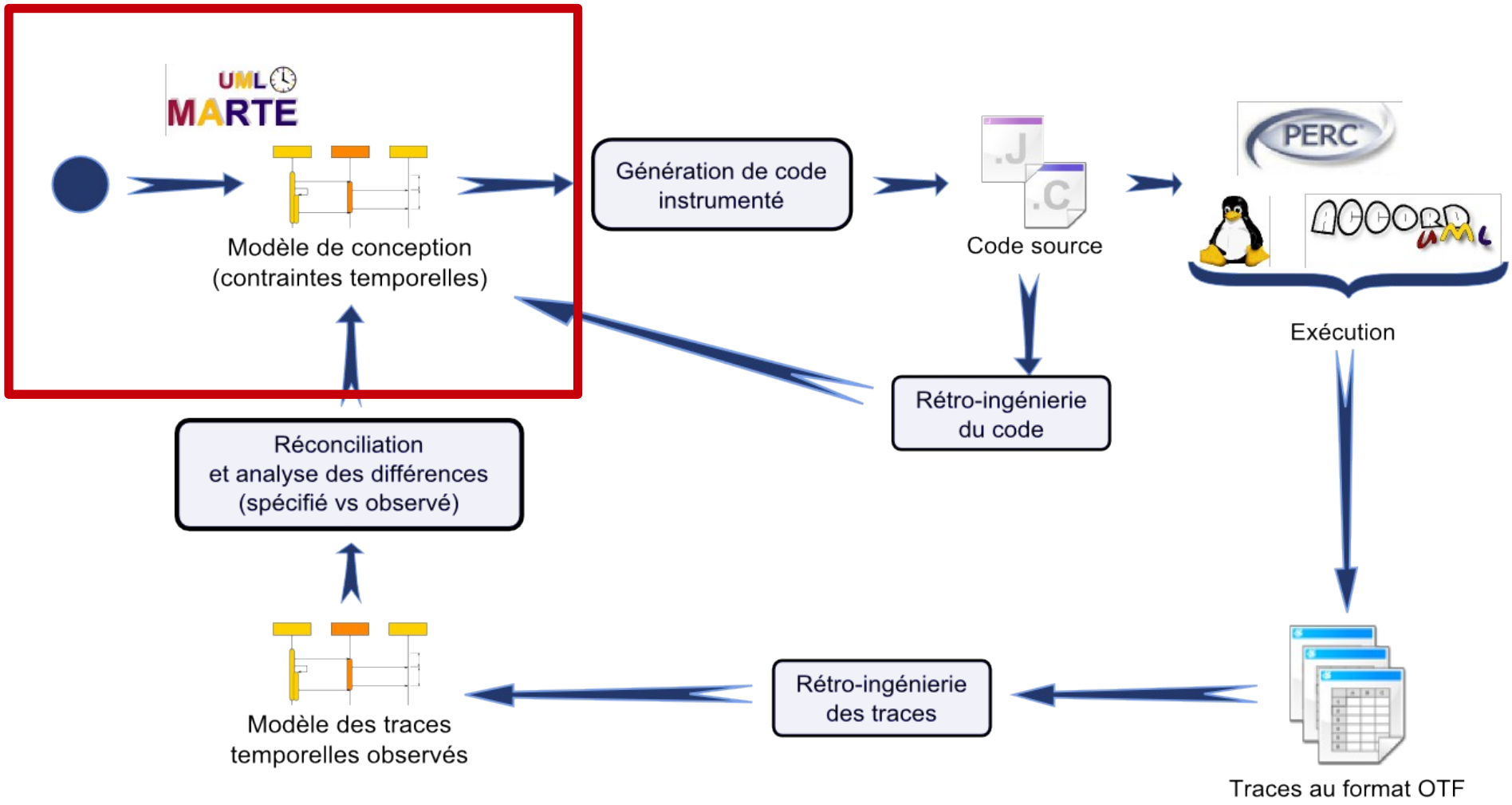
Check implementation traces



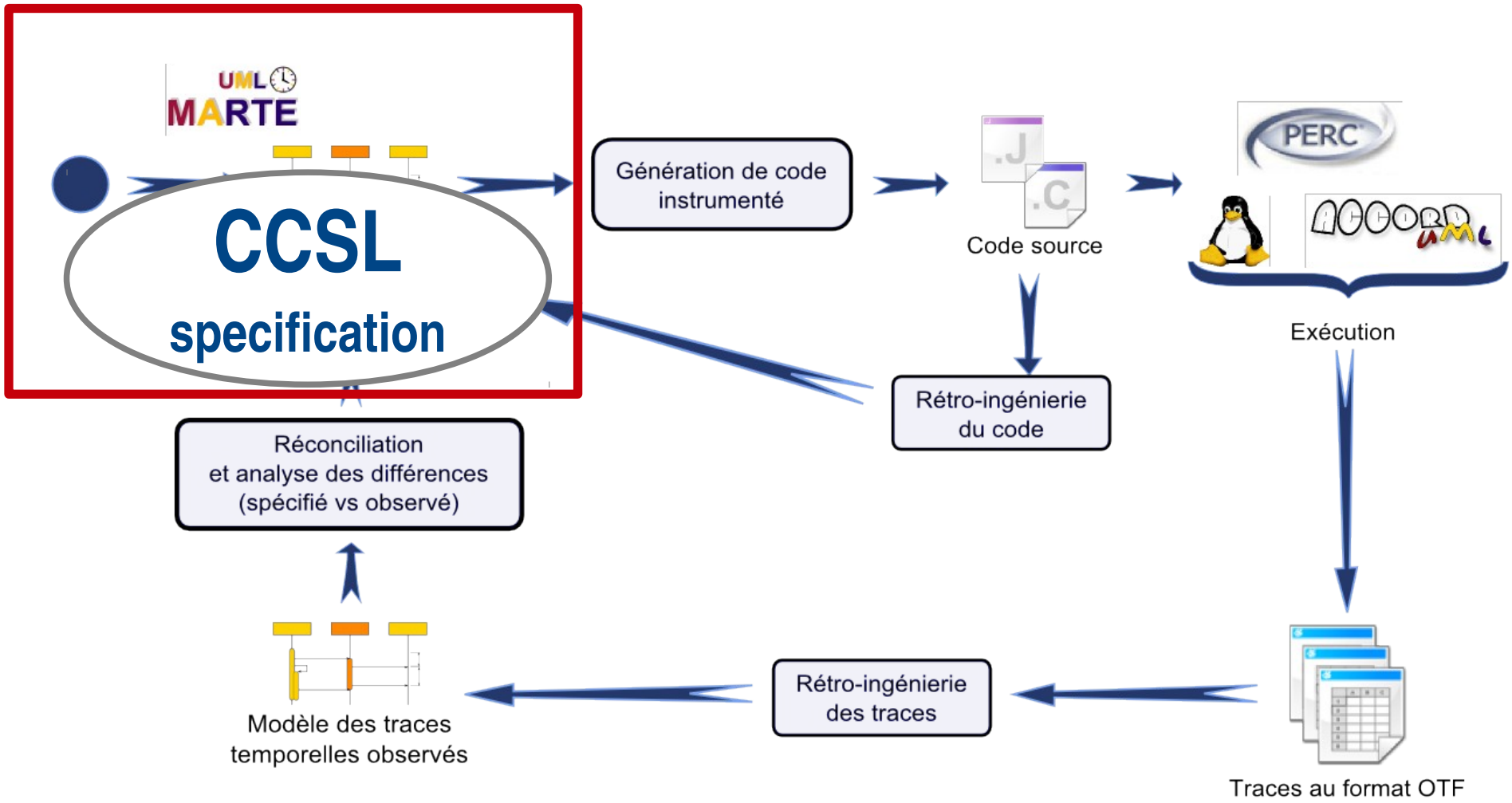
Processus RT-Simex



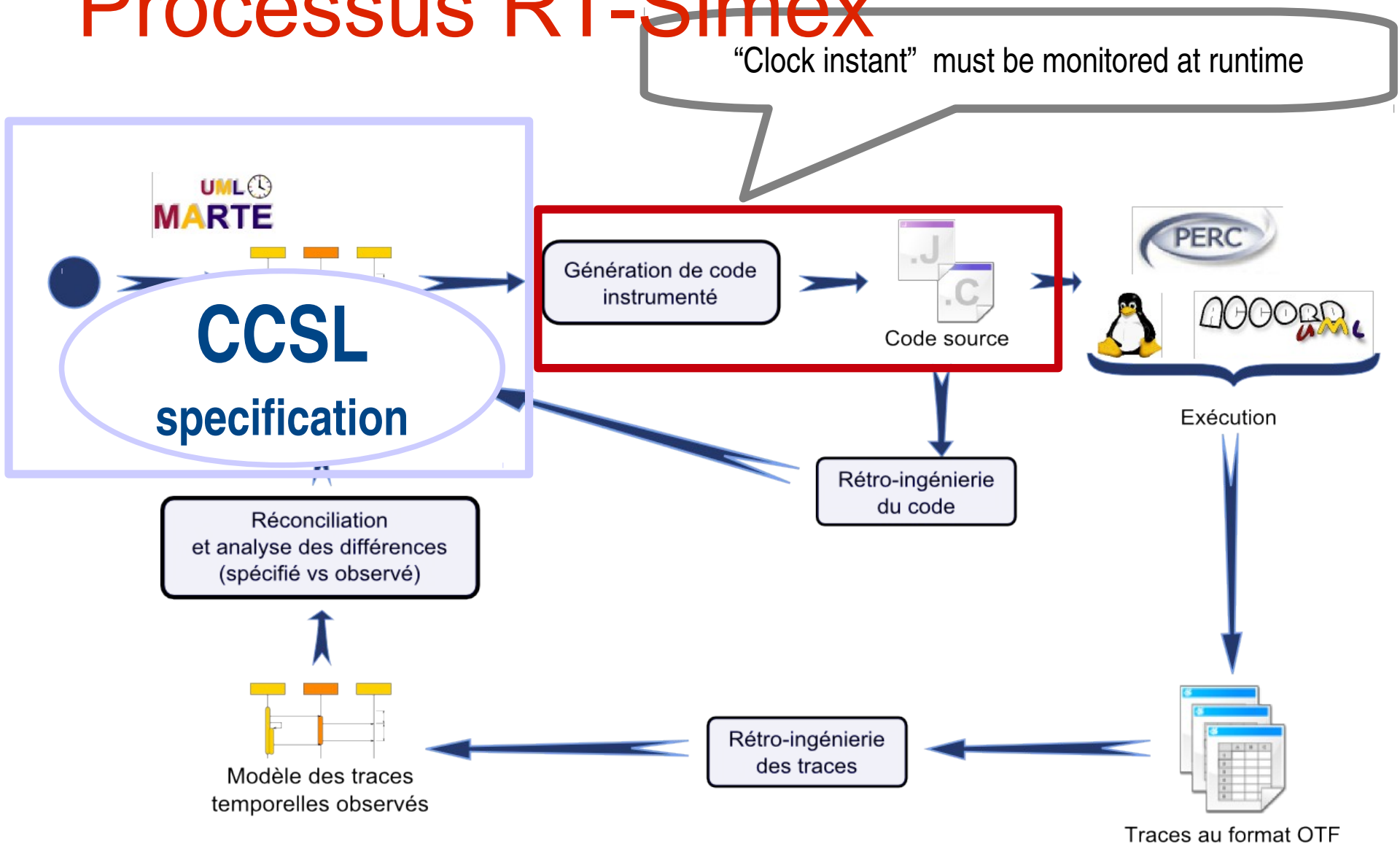
Processus RT-Simex



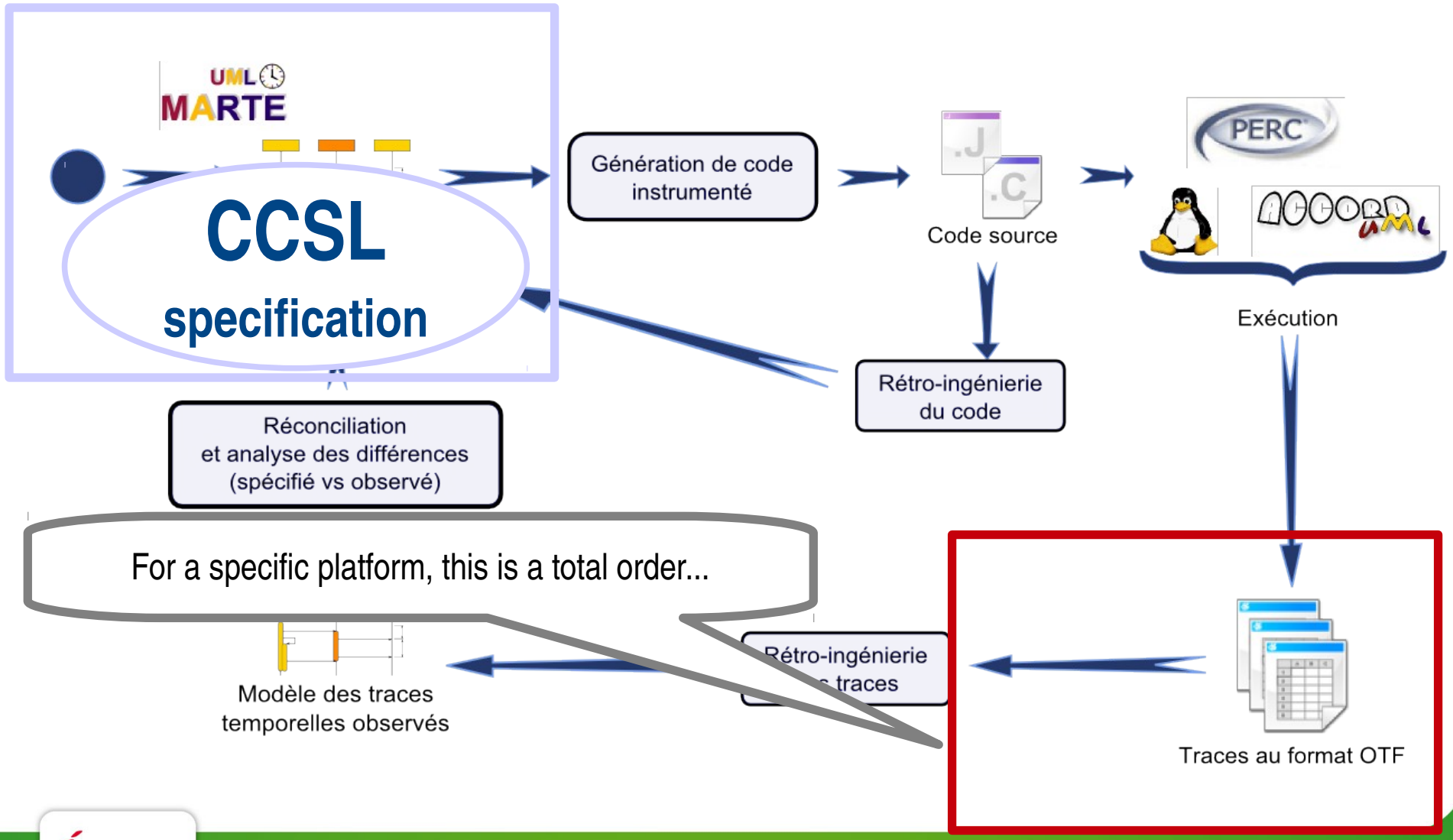
Processus RT-Simex



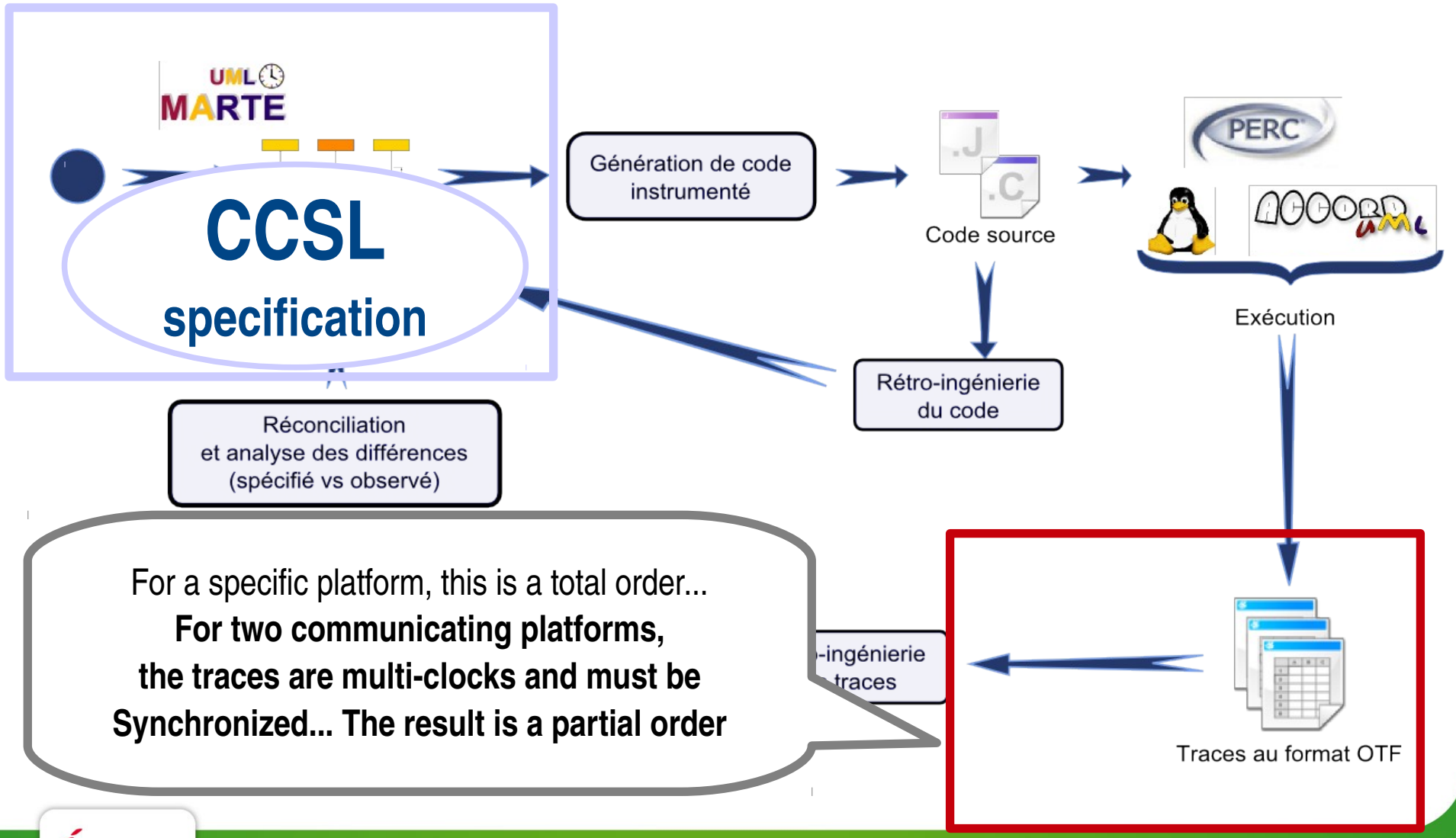
Processus RT-Simex



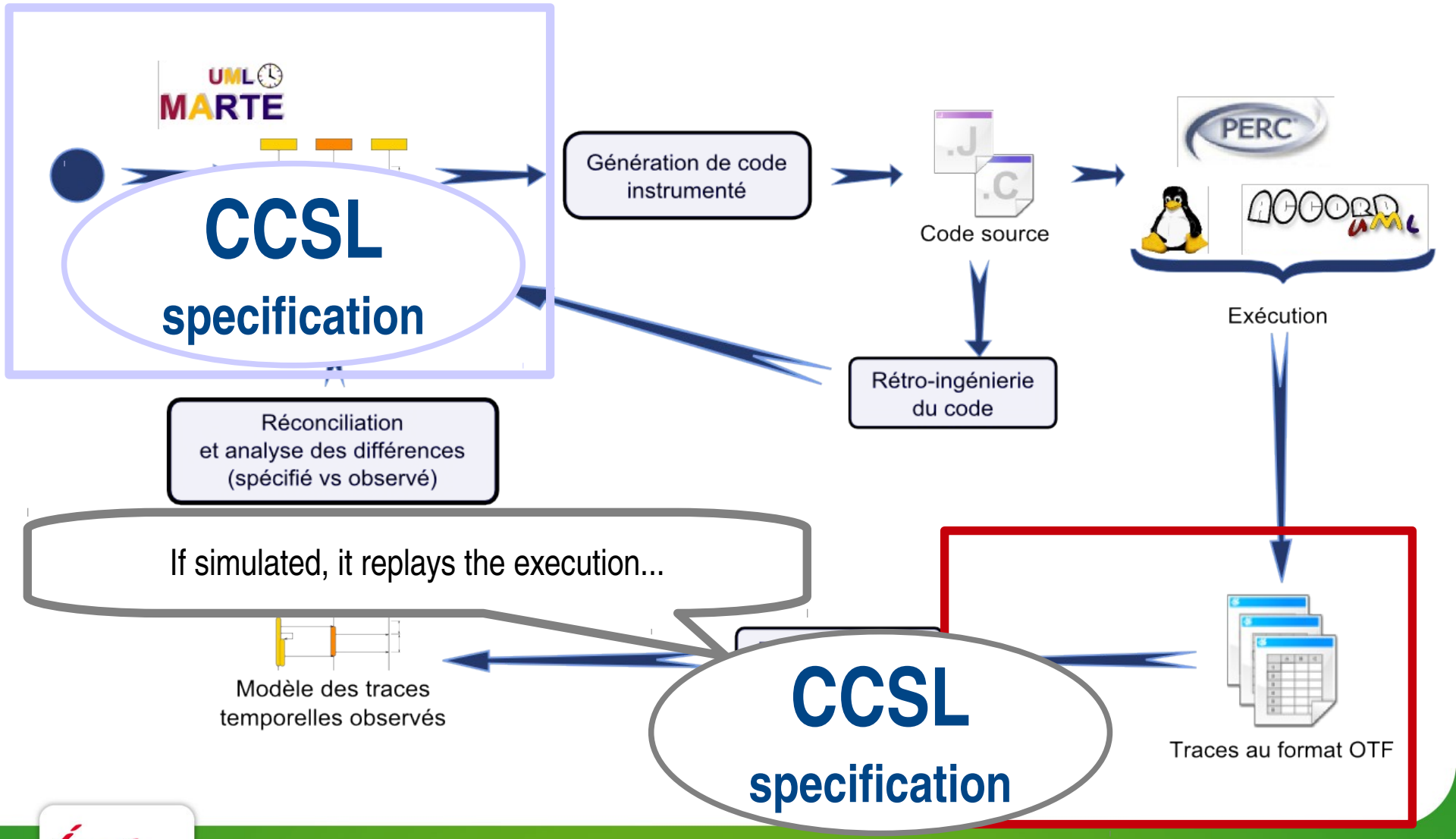
Processus RT-Simex



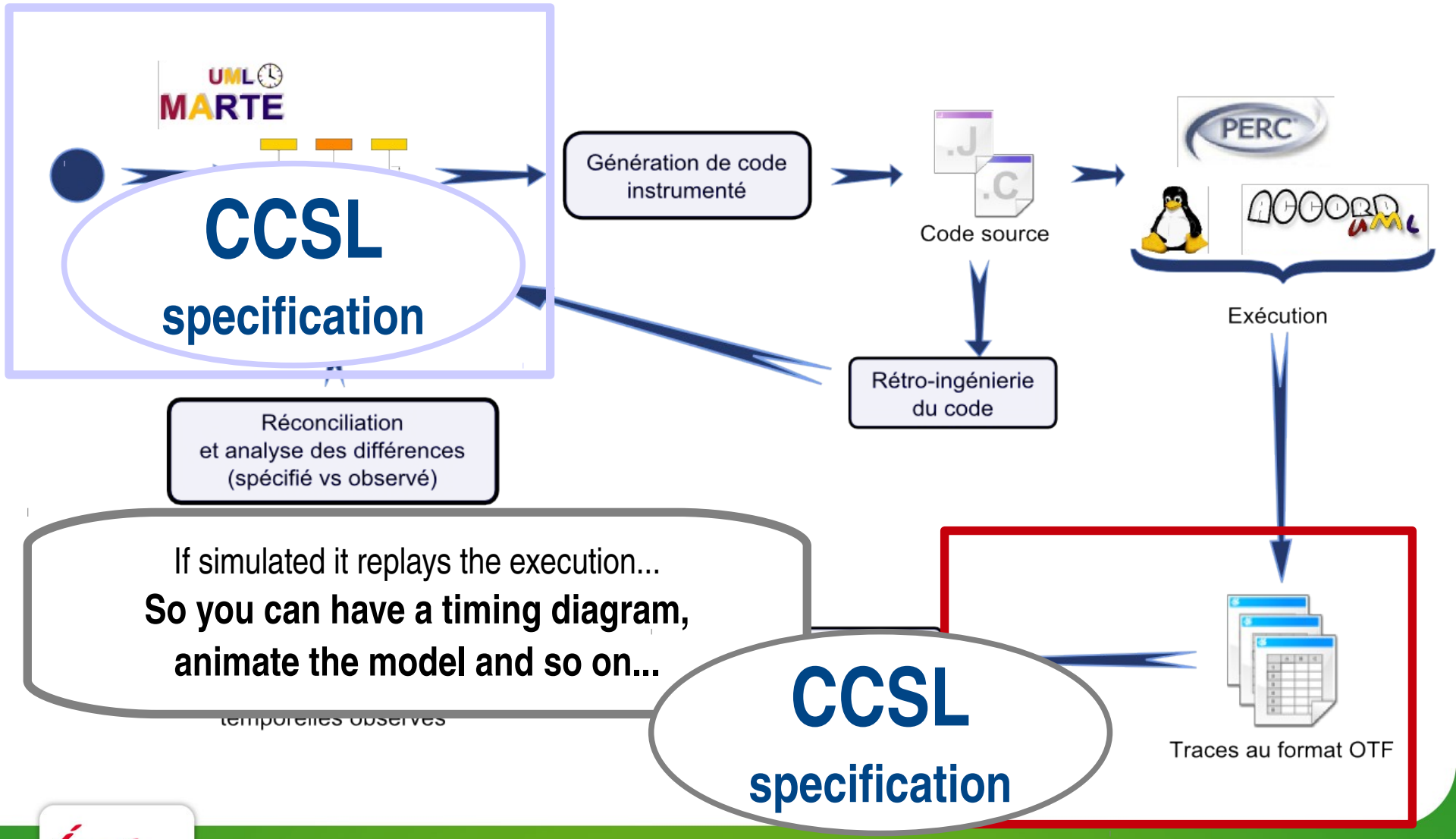
Processus RT-Simex



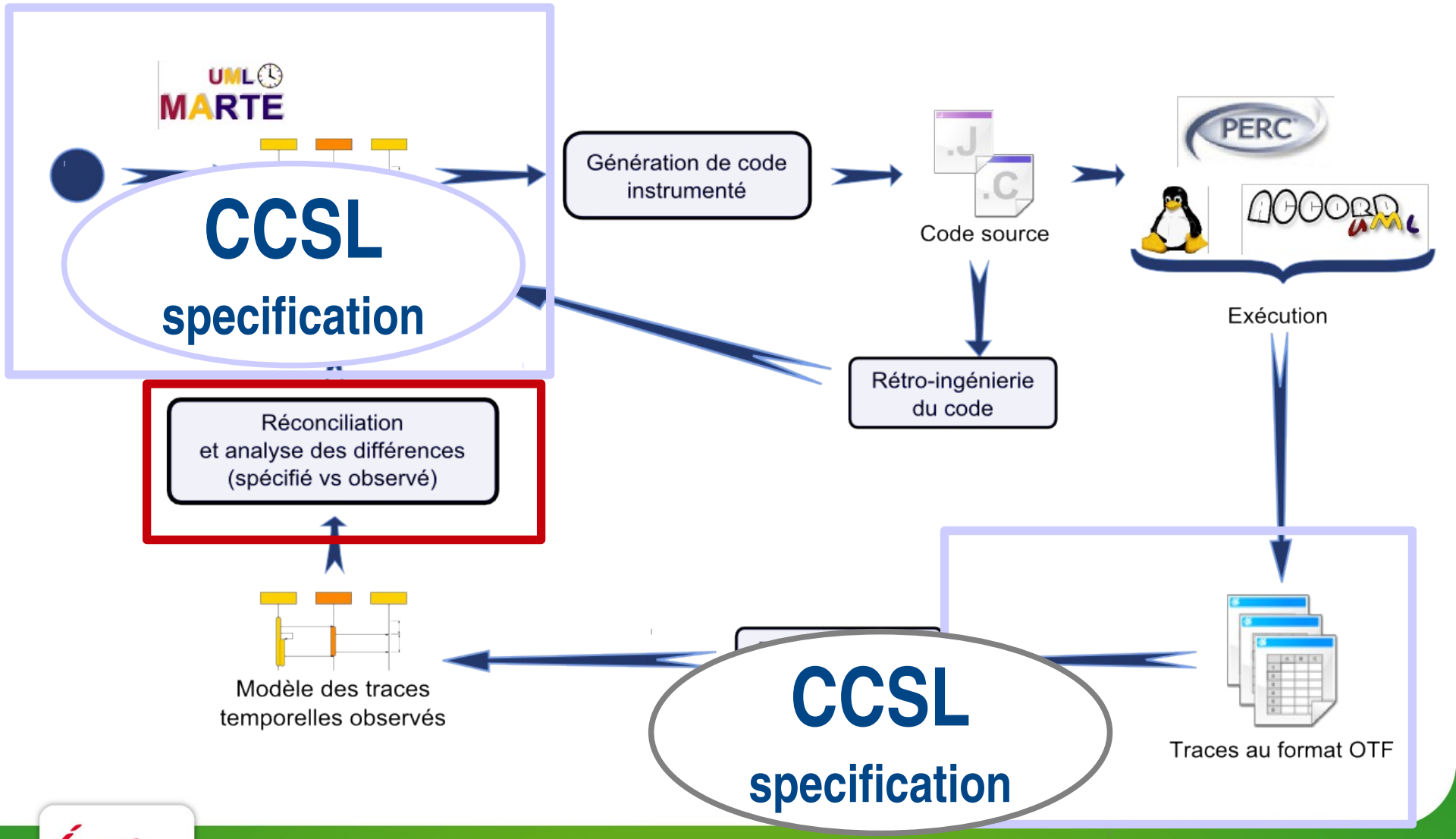
Processus RT-Simex



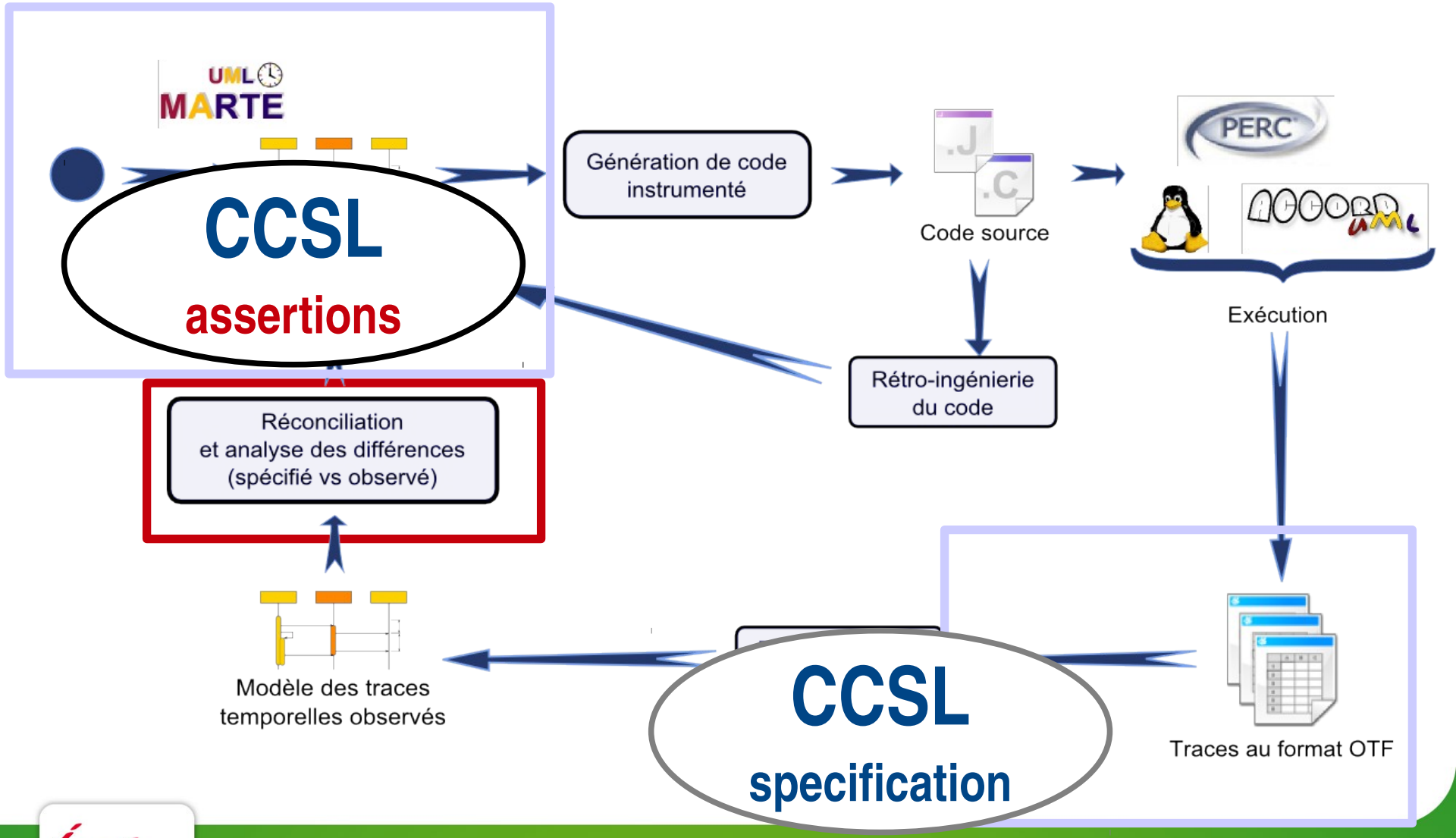
Processus RT-Simex



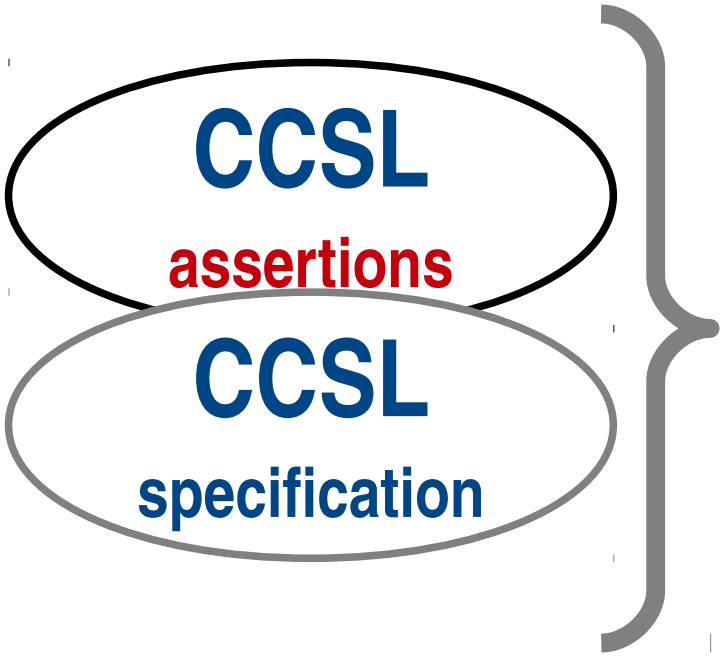
Processus RT-Simex



Processus RT-Simex

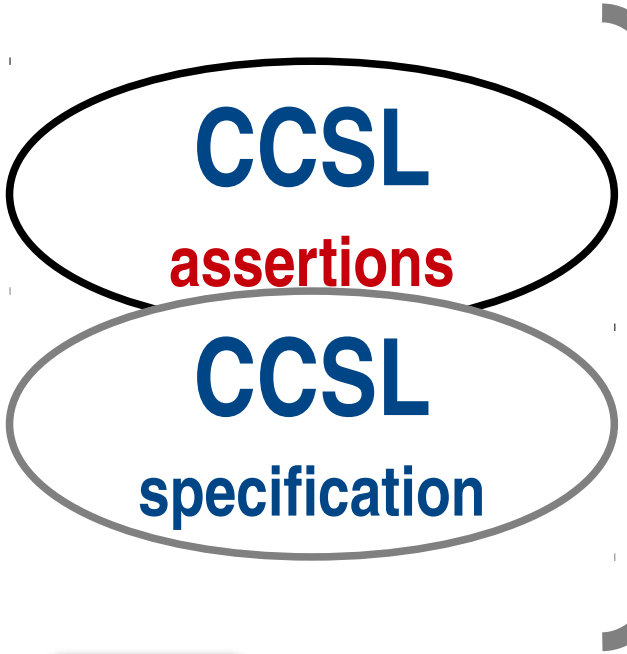


Processus RT-Simex

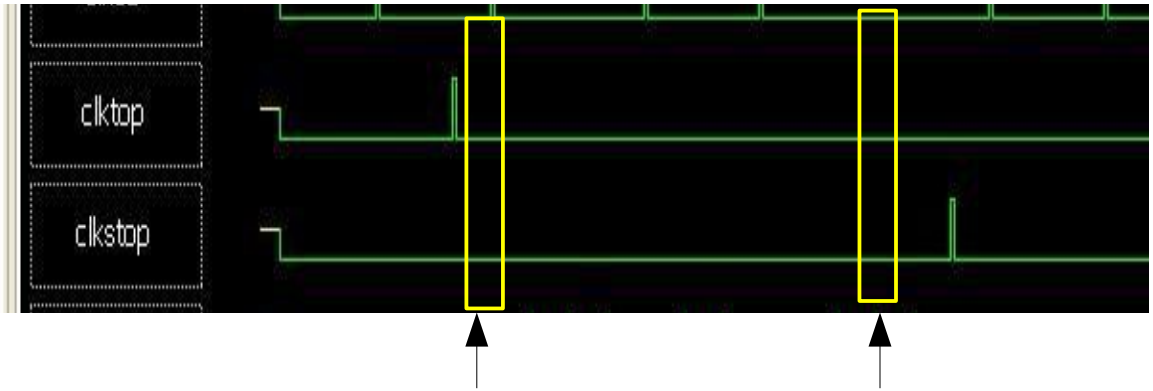


You can simulation both together to see if a violation occurs

Processus RT-Simex

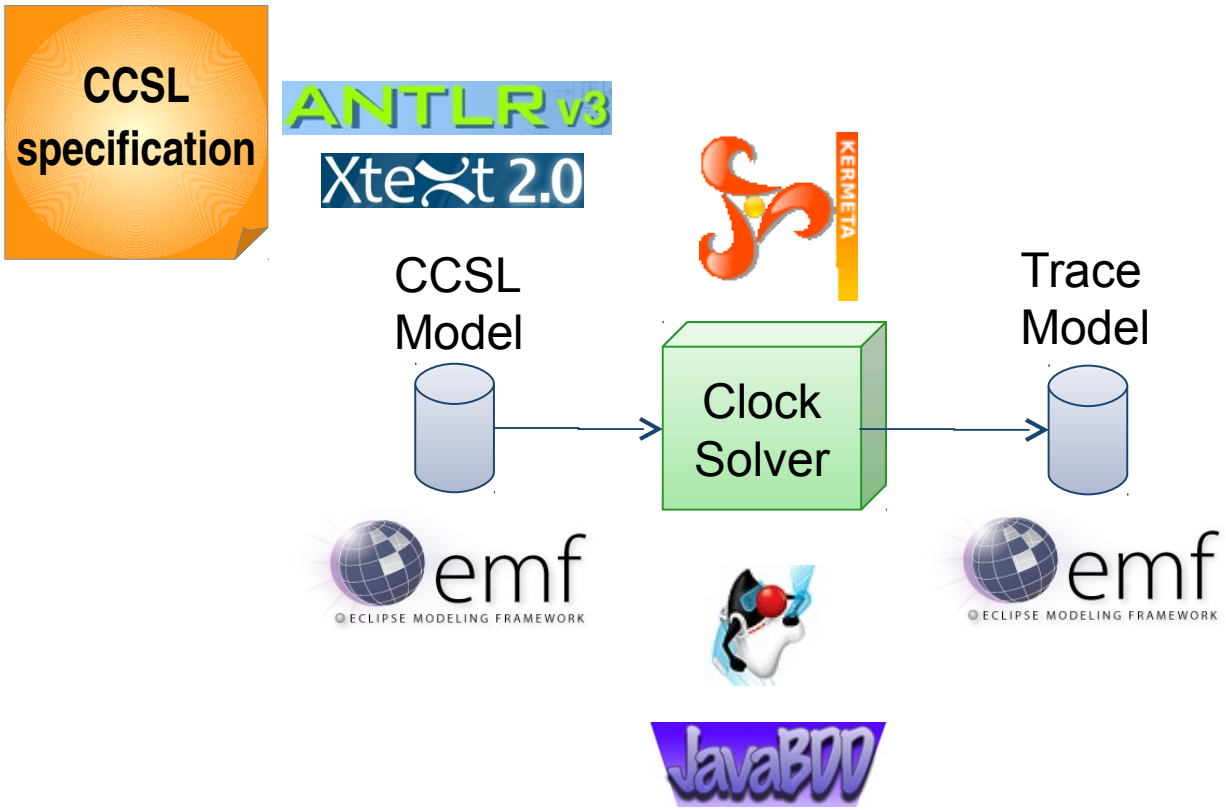


You can simulation both together to see if a violation occurs

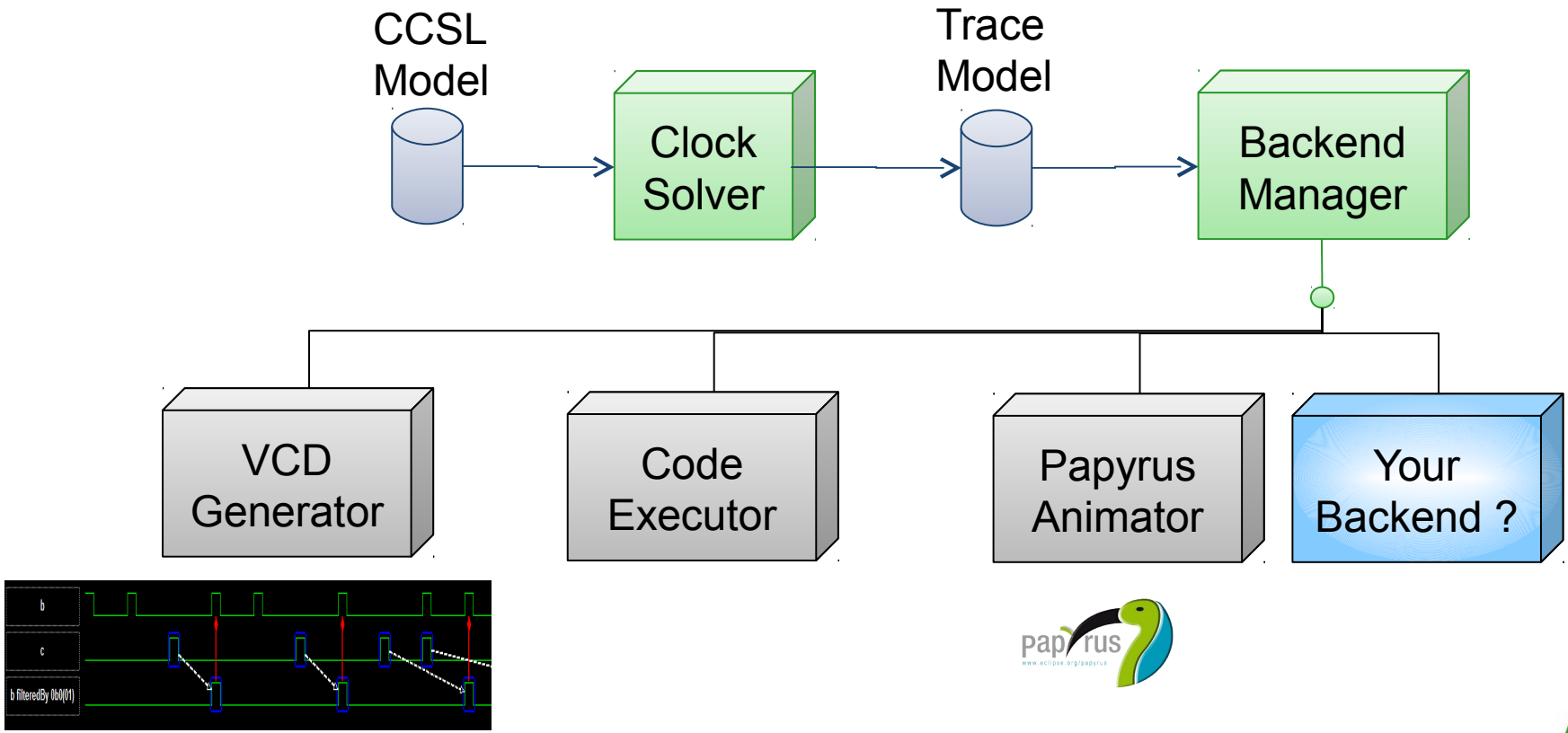


Violation of "Clk3b isSubClockOf clk3"

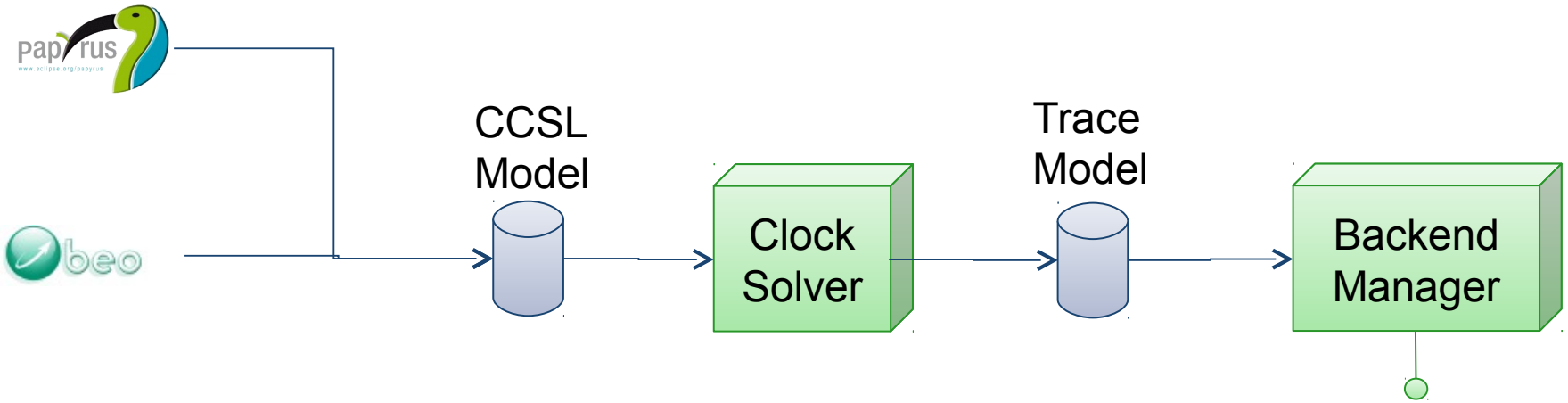
TimeSquare - Kernel



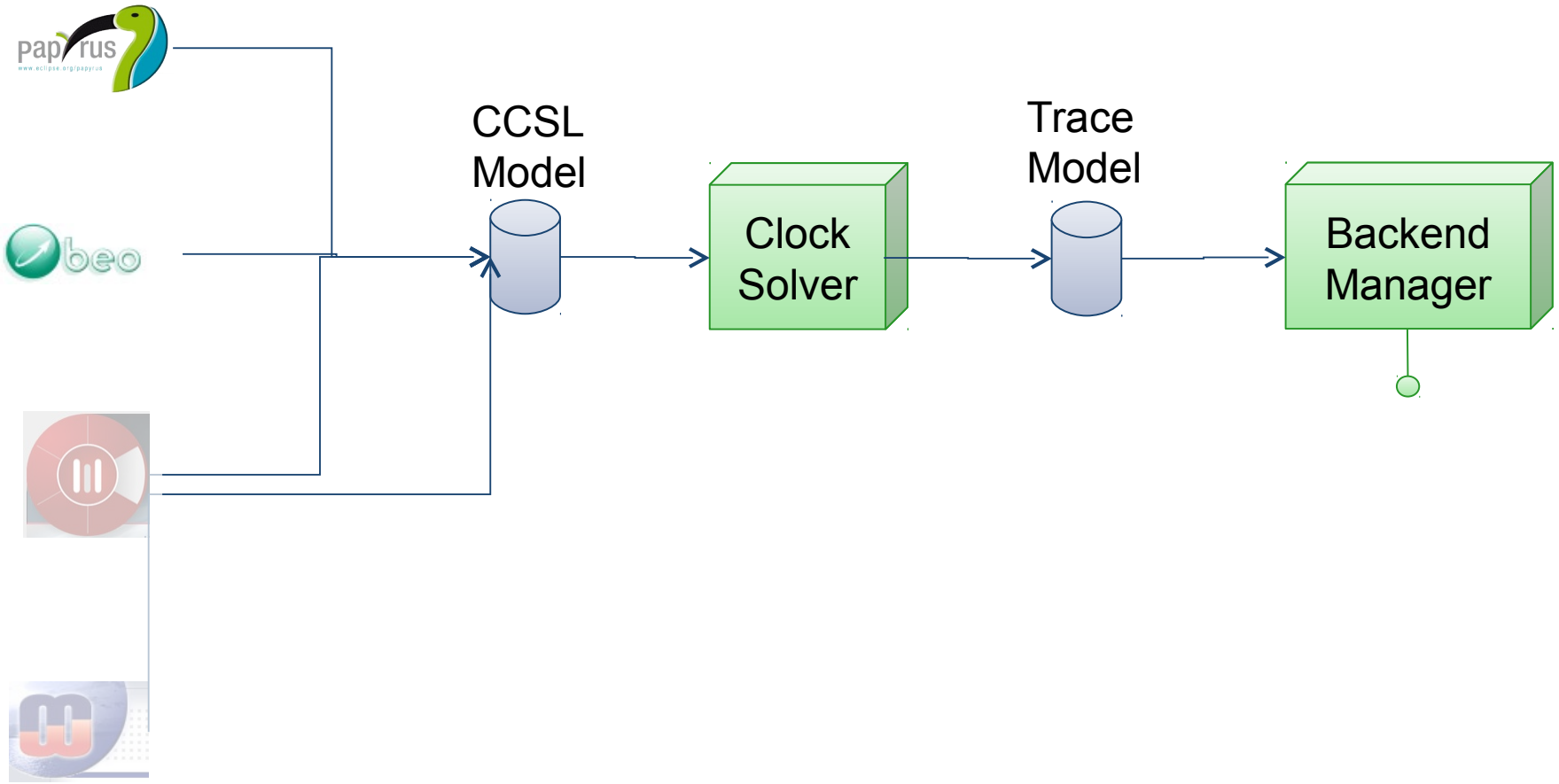
TimeSquare - Backend



TimeSquare - Frontend



TimeSquare - Frontend



4

Demonstration

Merci

Questions ?

timesquare.inria.fr