RMQFMU: Bridging the Real World with Co-simulation

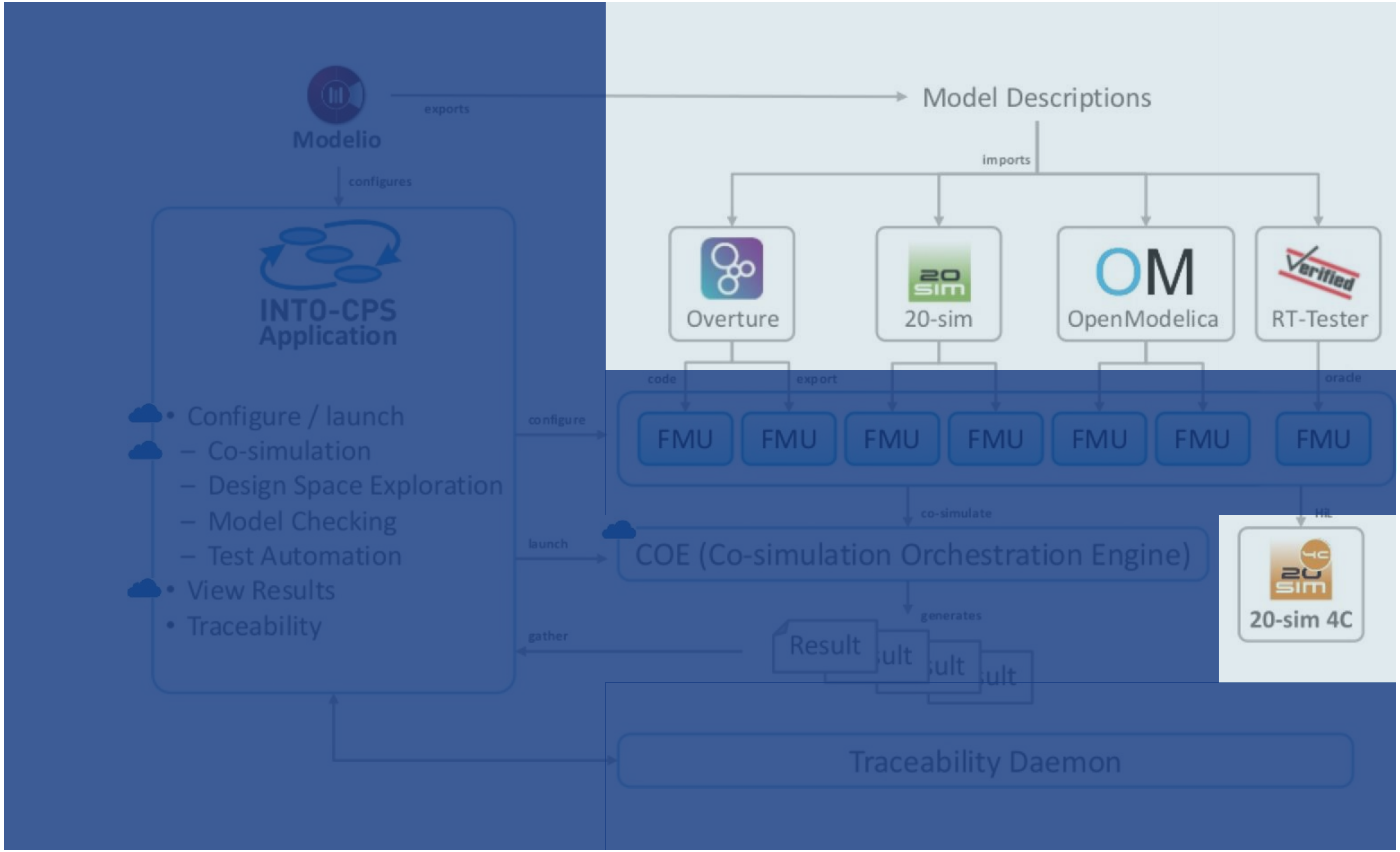
Mirgita Frasheri, Henrik Ejersbo, Casper Thule, and Lukas Esterle

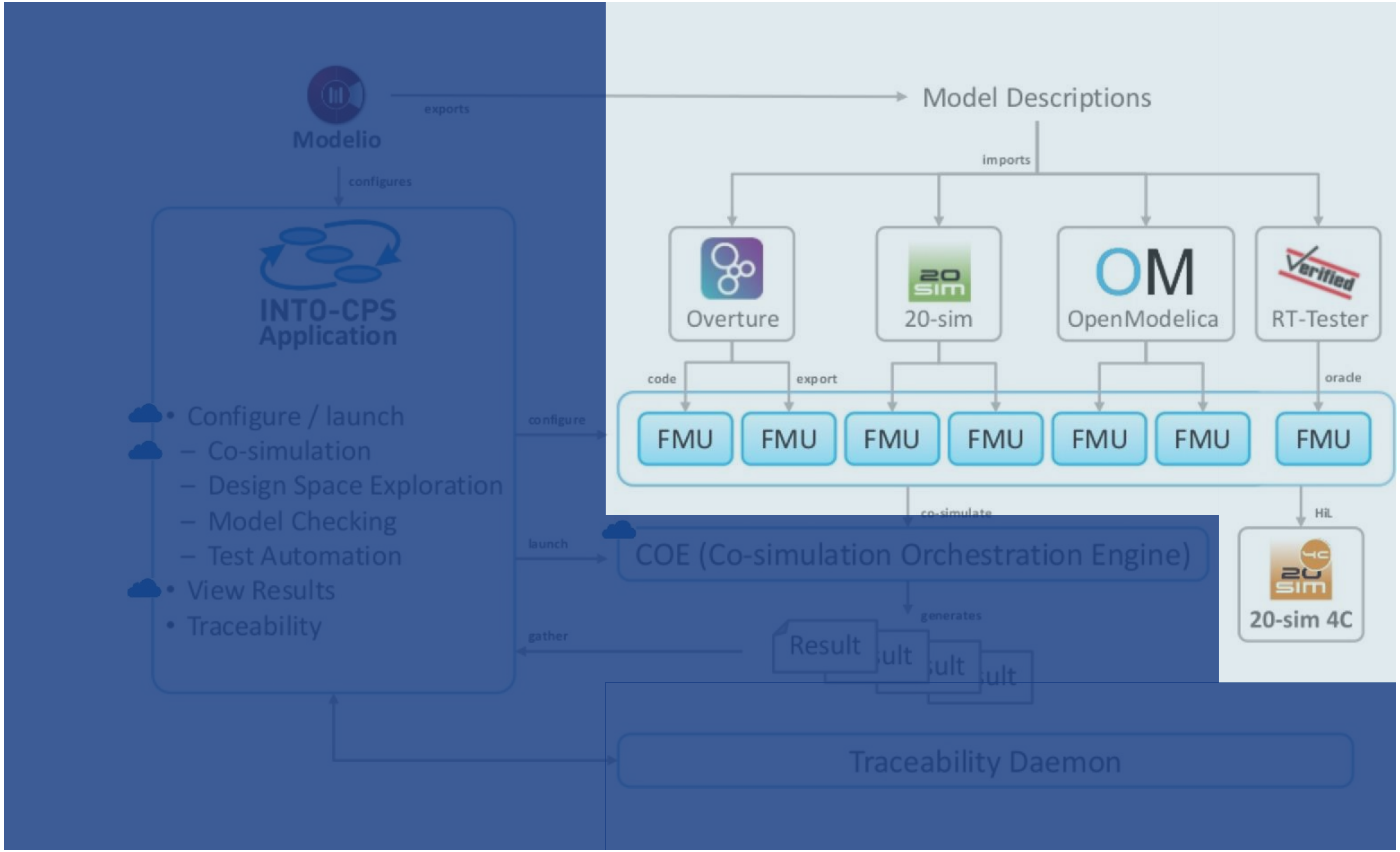
Overture Workshop

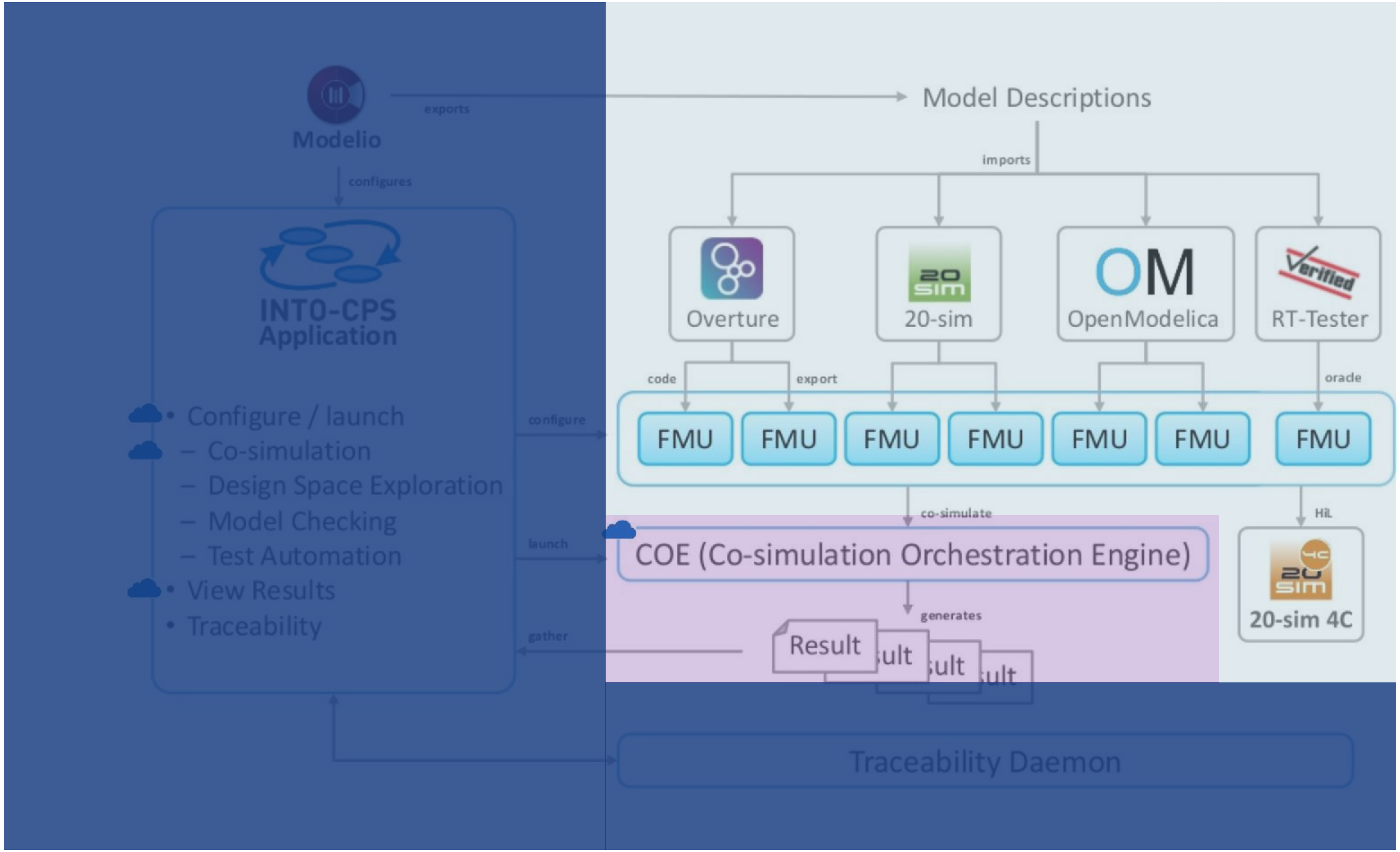
22nd October 2021

RMQFMU

RMQFMU


Modelio
exports
Model Descriptions
imports
configures
INTO-CPS Application
• Configure / launch
– Co-simulation
– Design Space Exploration
– Model Checking
– Test Automation
• View Results
• Traceability
Overture
20-sim
OpenModelica
RT-Tester
code
export
oracle
configure
FMU
FMU
FMU
FMU
FMU
FMU
FMU
co-simulate
HIL
launch
COE (Co-simulation Orchestration Engine)
20-sim 4C
generates
gather
Result
Result
Result
Result
Traceability Daemon


Modelio
exports
Model Descriptions
imports
configures
INTO-CPS
Application
Overture
20-sim
OpenModelica
RT-Tester
code
export
oracle
• Configure / launch
– Co-simulation
– Design Space Exploration
– Model Checking
– Test Automation
• View Results
• Traceability
configure
FMU
FMU
FMU
FMU
FMU
FMU
FMU
co-simulate
HiL
launch
COE (Co-simulation Orchestration Engine)
20-sim 4C
generates
gather
Result
Result
Result
Result
Traceability Daemon

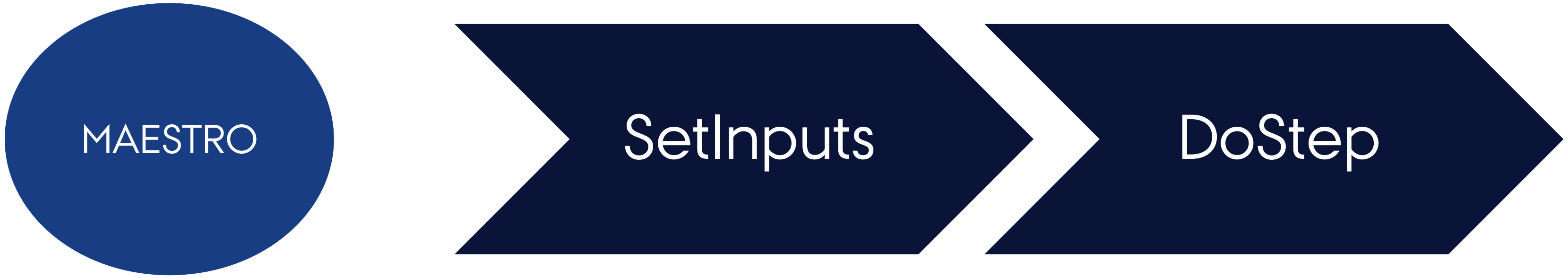


Modelio
exports
Model Descriptions
imports
configures
INTO-CPS Application
Configure / launch
– Co-simulation
– Design Space Exploration
– Model Checking
– Test Automation
View Results
Traceability
Overture
20-sim
OpenModelica
RT-Tester
code
export
oracle
configure
FMU
FMU
FMU
FMU
FMU
FMU
FMU
co-simulate
HiL
launch
COE (Co-simulation Orchestration Engine)
20-sim 4C
generates
gather
Result
Traceability Daemon

MAESTRO

MAESTRO

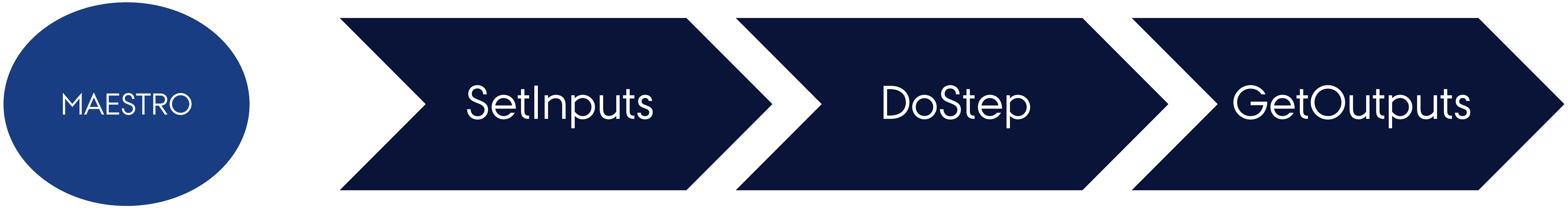
SetInputs

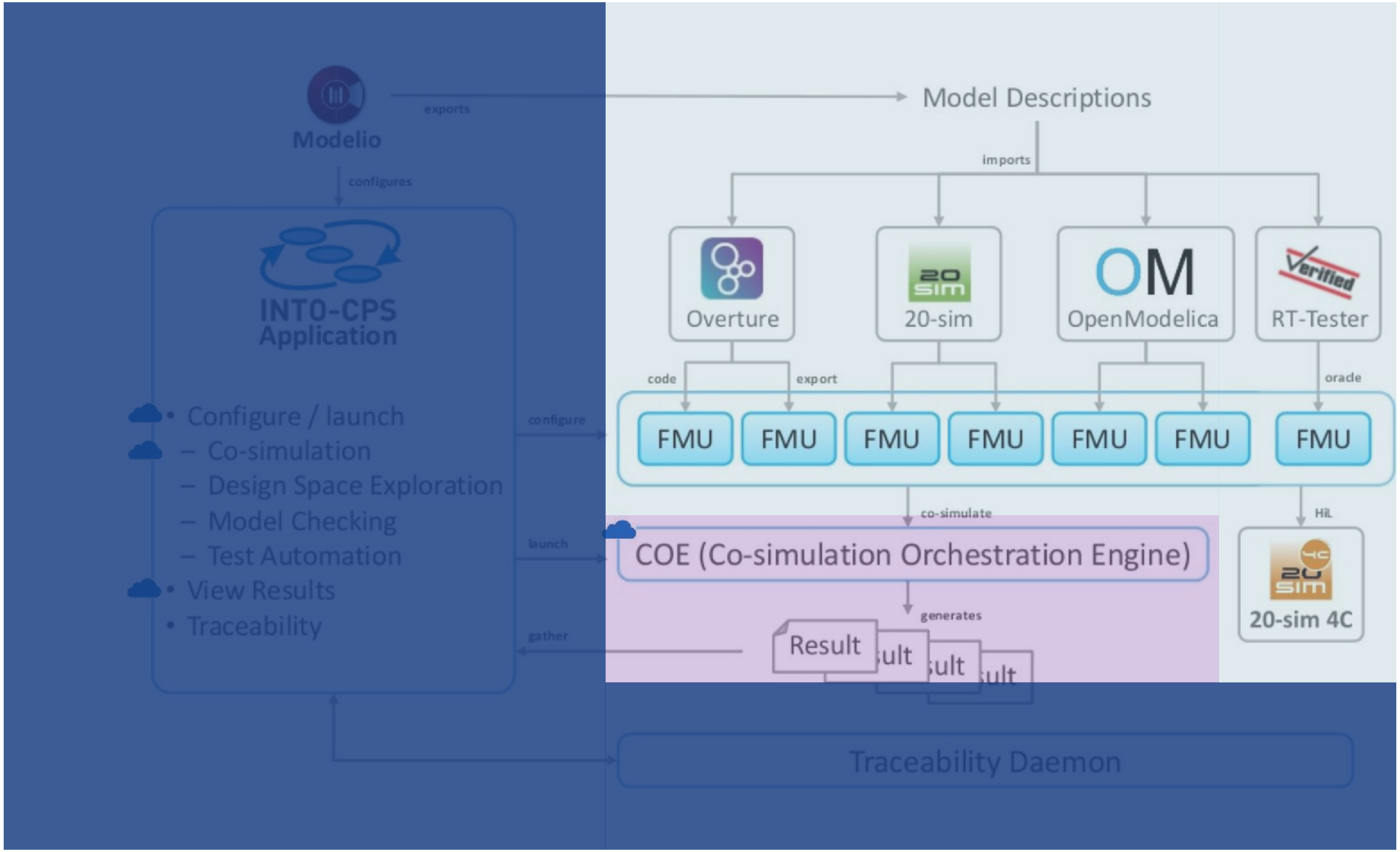


MAESTRO

SetInputs

DoStep


MAESTRO
SetInputs
DoStep
GetOutputs



Modelio

exports

Model Descriptions

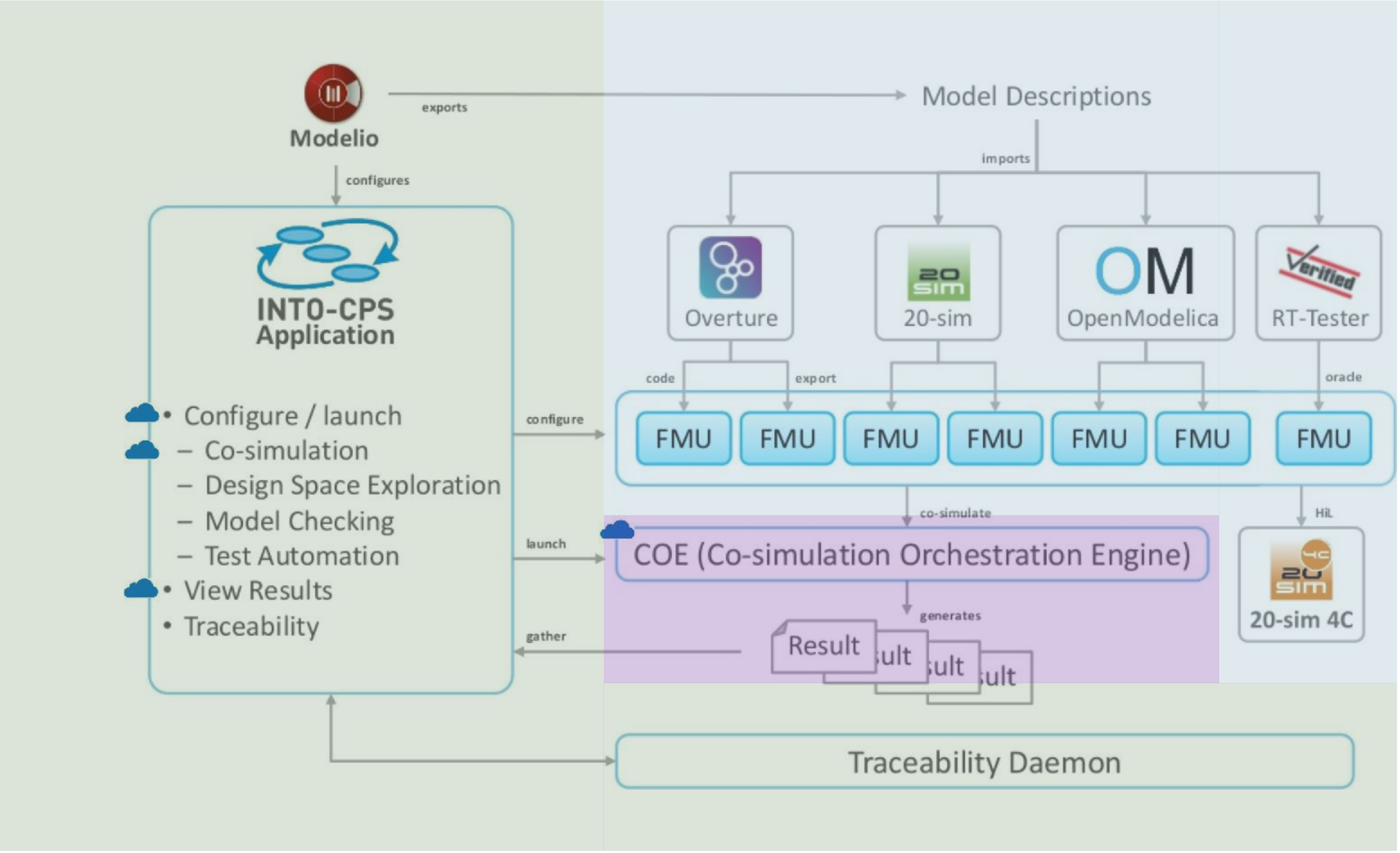
configures

INTO-CPS Application

- Configure / launch
  - Co-simulation
  - Design Space Exploration
  - Model Checking
  - Test Automation
- View Results
- Traceability

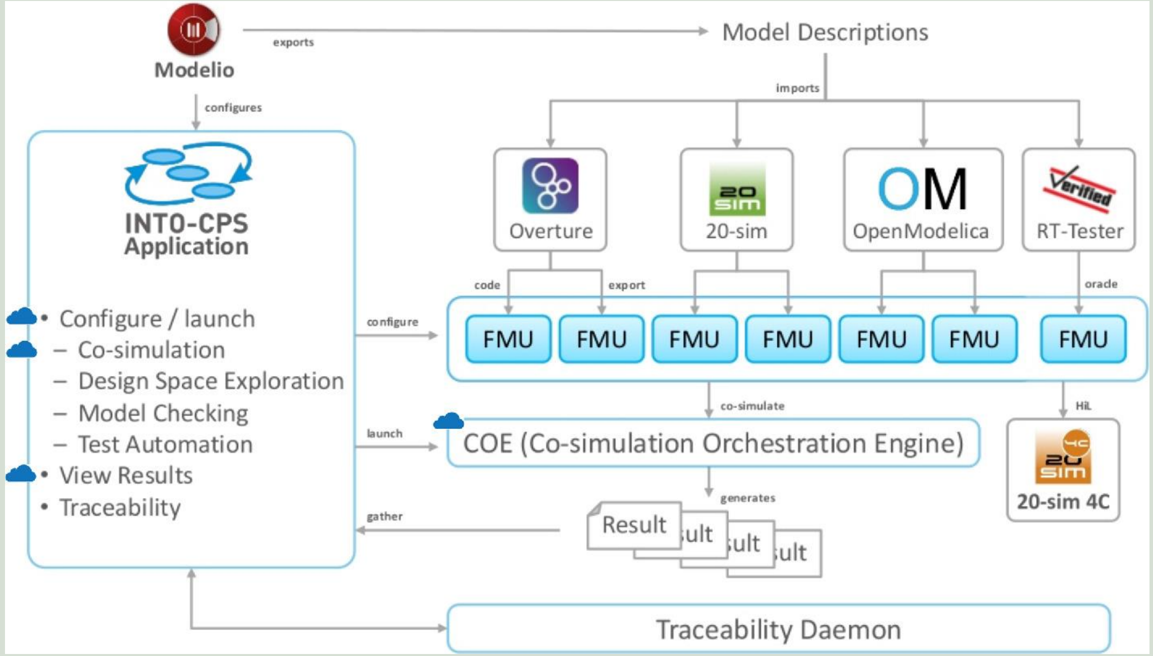
COE (Co-simulation Orchestration Engine)

Traceability Daemon


Modelio
exports
Model Descriptions
imports
configures
INTO-CPS Application
• Configure / launch
– Co-simulation
– Design Space Exploration
– Model Checking
– Test Automation
• View Results
• Traceability
Overture
20-sim
OpenModelica
RT-Tester
code
export
oracle
configure
FMU
FMU
FMU
FMU
FMU
FMU
FMU
co-simulate
HiL
launch
COE (Co-simulation Orchestration Engine)
20-sim 4C
generates
gather
Result
Traceability Daemon

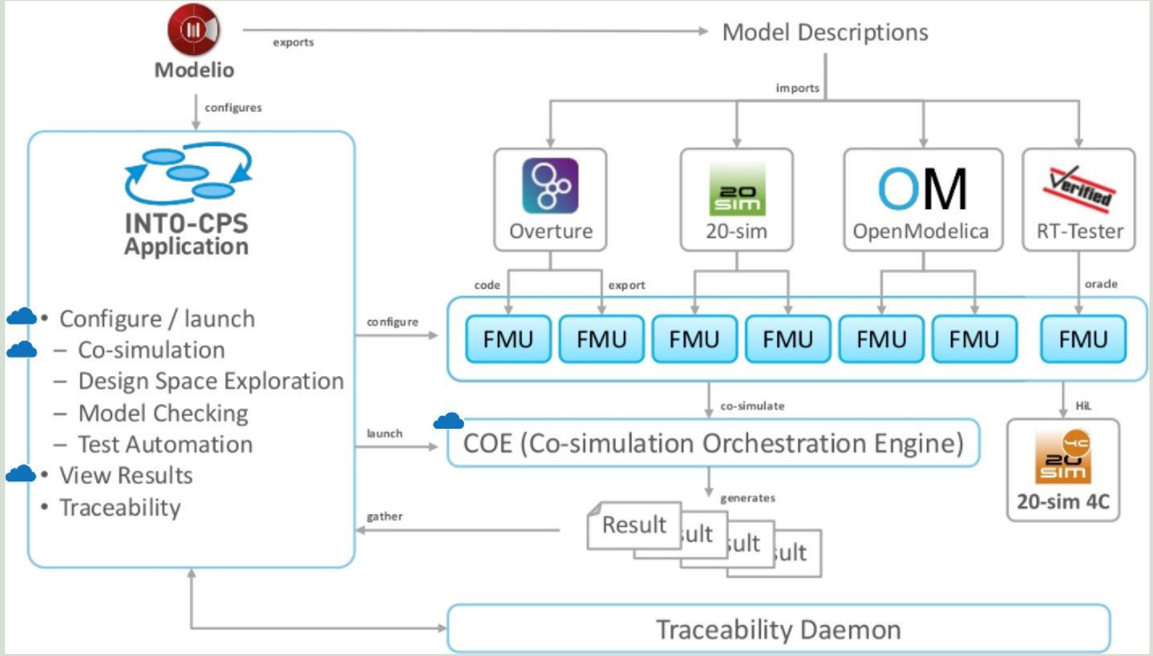
RMQFMU

Co-Simulation World



- Configure / launch
  - Co-simulation
  - Design Space Exploration
  - Model Checking
  - Test Automation
- View Results
- Traceability

Co-Simulation World



'Real' World

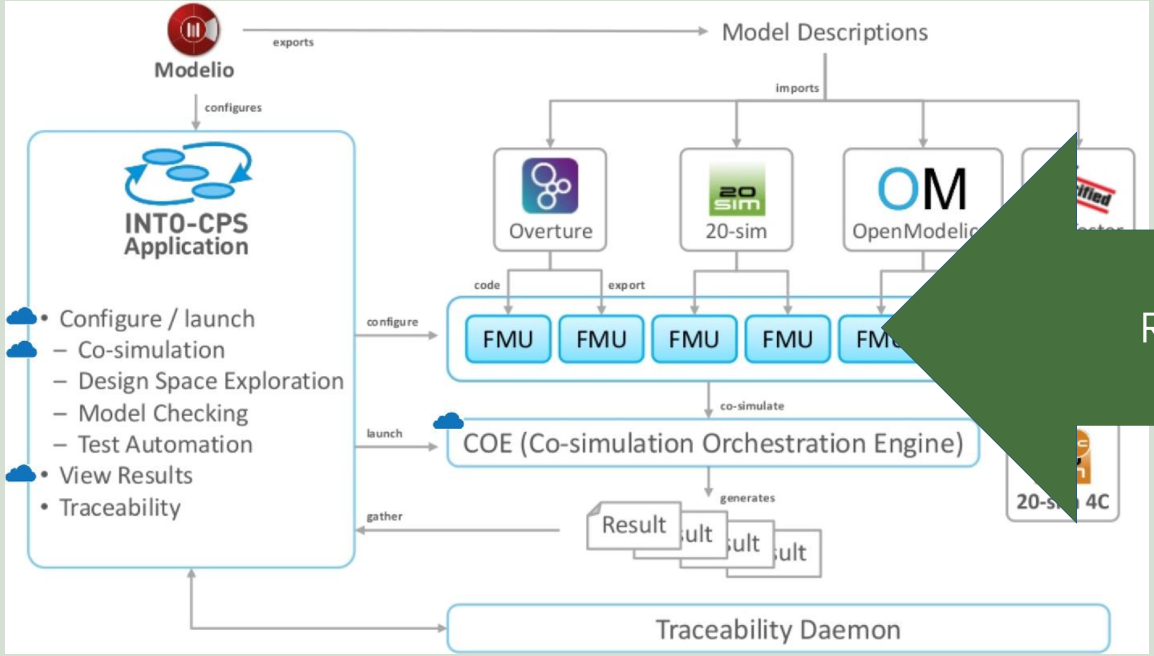
A Robot

ROBOTTI
ROBOTTI

A Gazebo Simulation

Co-Simulation World

'Real' World



RMQFMU

A Robot

A Gazebo Simulation

Features

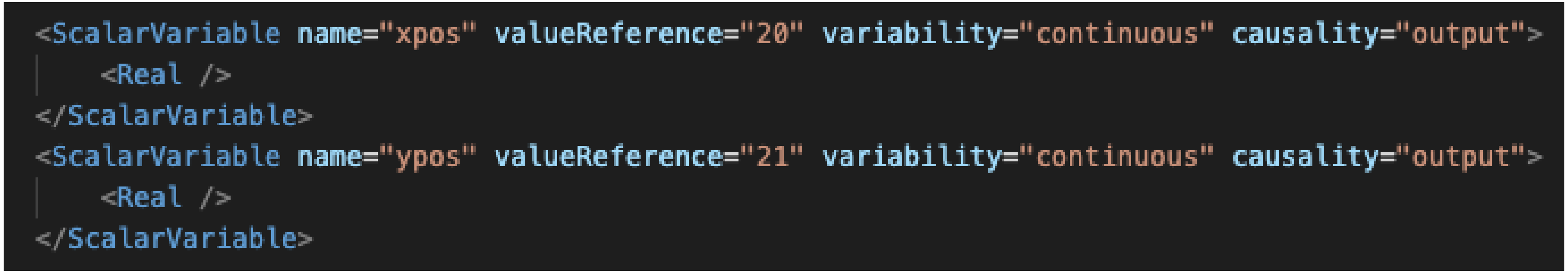
- Receive data

Features

- Receive data
    - Specify outputs in the *ModelDescription.xml*

Features

- Receive data
  - Specify outputs in the *ModelDescription.xml*
  - E.g. receive (x,y) positions of robot



```
<ScalarVariable name="xpos" valueReference="20" variability="continuous" causality="output">
    <Real />
</ScalarVariable>
<ScalarVariable name="ypos" valueReference="21" variability="continuous" causality="output">
    <Real />
</ScalarVariable>
```

- Receive data
  - Specify outputs in the *ModelDescription.xml*
  - E.g. receive (x,y) positions of robot
  - Quality attributes: *maxage* and *lookahead*

Features

➢ Receive data

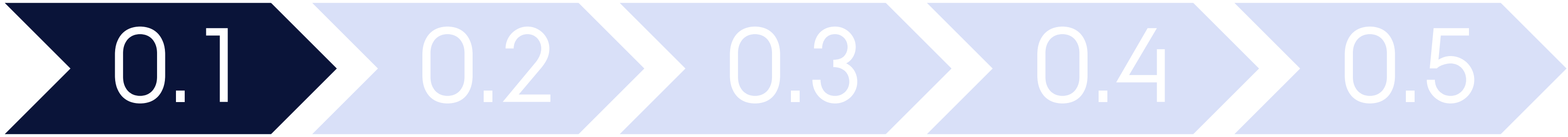
- Specify outputs in the *ModelDescription.xml*
- E.g. receive (x,y) positions of robot
- Quality attributes: *maxage* and *lookahead*

0.1 0.2 0.3 0.4 0.5

Features

- Receive data
  - Specify outputs in the *ModelDescription.xml*
  - E.g. receive (x,y) positions of robot
  - Quality attributes: *maxage* and *lookahead*

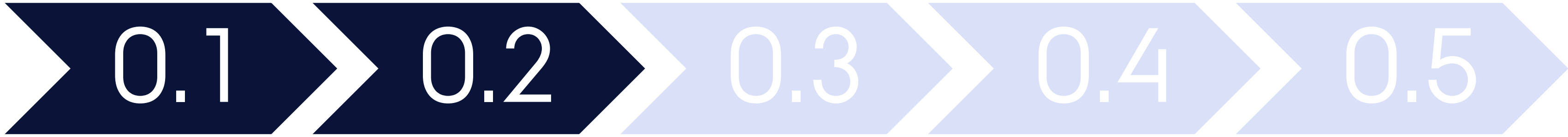
ma = 200ms



Features

- Receive data
  - Specify outputs in the *ModelDescription.xml*
  - E.g. receive (x,y) positions of robot
  - Quality attributes: *maxage* and *lookahead*

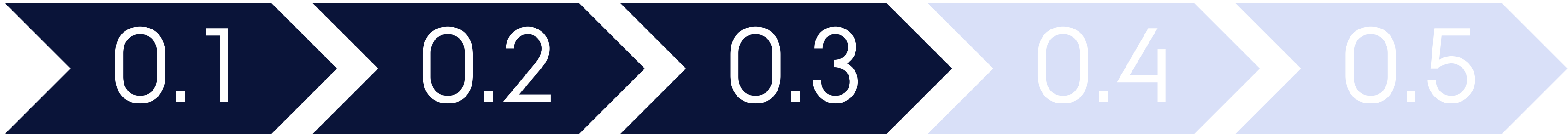
ma = 200ms



0.1 0.2 0.3 0.4 0.5

- Receive data
  - Specify outputs in the *ModelDescription.xml*
  - E.g. receive (x,y) positions of robot
  - Quality attributes: *maxage* and *lookahead*

ma = 200ms

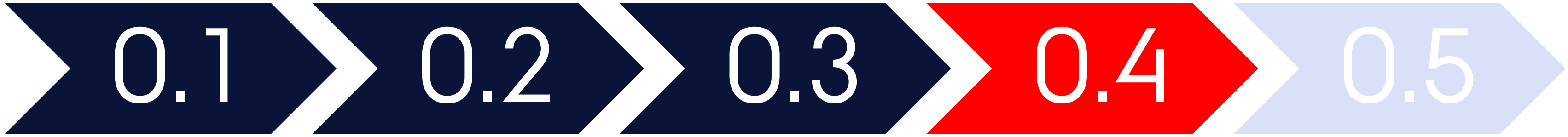


0.1 0.2 0.3 0.4 0.5

Receive data

- Specify outputs in the *ModelDescription.xml*
- E.g. receive (x,y) positions of robot
- Quality attributes: *maxage* and *lookahead*

ma = 200ms



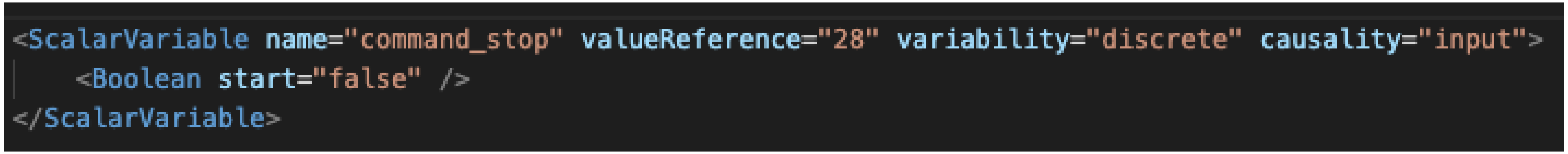
- Receive data
  - Specify outputs in the *ModelDescription.xml*
  - E.g. receive (x,y) positions of robot
  - Quality attributes: *maxage* and *lookahead*

- Send data
  - Specify inputs in the *ModelDescription.xml*

Features

- Receive data
  - Specify outputs in the *ModelDescription.xml*
  - E.g. receive (x,y) positions of robot
  - Quality attributes: *maxage* and *lookahead*

- Send data
  - Specify inputs in the *ModelDescription.xml*
  - E.g. send a stop command to the robot



```
<ScalarVariable name="command_stop" valueReference="28" variability="discrete" causality="input">
    <Boolean start="false" />
</ScalarVariable>
```

Features

- Receive data
  - Specify outputs in the *ModelDescription.xml*
  - E.g. receive (x,y) positions of robot
  - Quality attributes: *maxage* and *lookahead*

- Send data
  - Specify inputs in the *ModelDescription.xml*
  - E.g. send a stop command to the robot
  - Send on change

Features

Features

- Receive data
  - Specify outputs in the *ModelDescription.xml*
  - E.g. receive (x,y) positions of robot
  - Quality attributes: *maxage* and *lookahead*

- Send data
  - Specify inputs in the *ModelDescription.xml*
  - E.g. send a stop command to the robot
  - Send on change

- Threaded operation
  - De-couple the *doStep* call from consuming from RabbitMQ

Evaluation

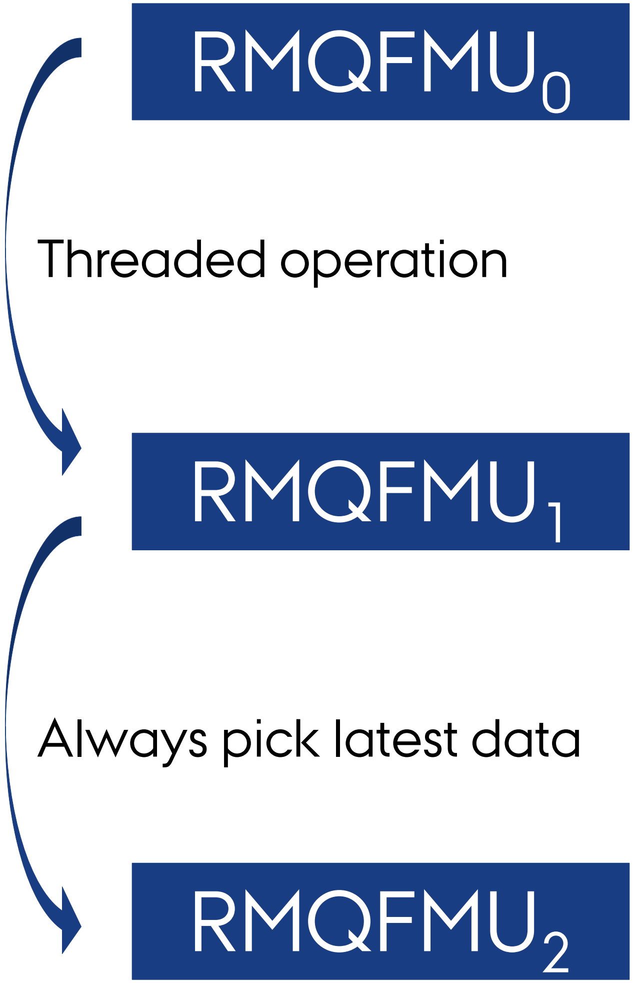
RMQFMU$_0$

RMQFMU$_0$

Threaded operation

RMQFMU$_1$

Evaluation



RMQFMU$_0$

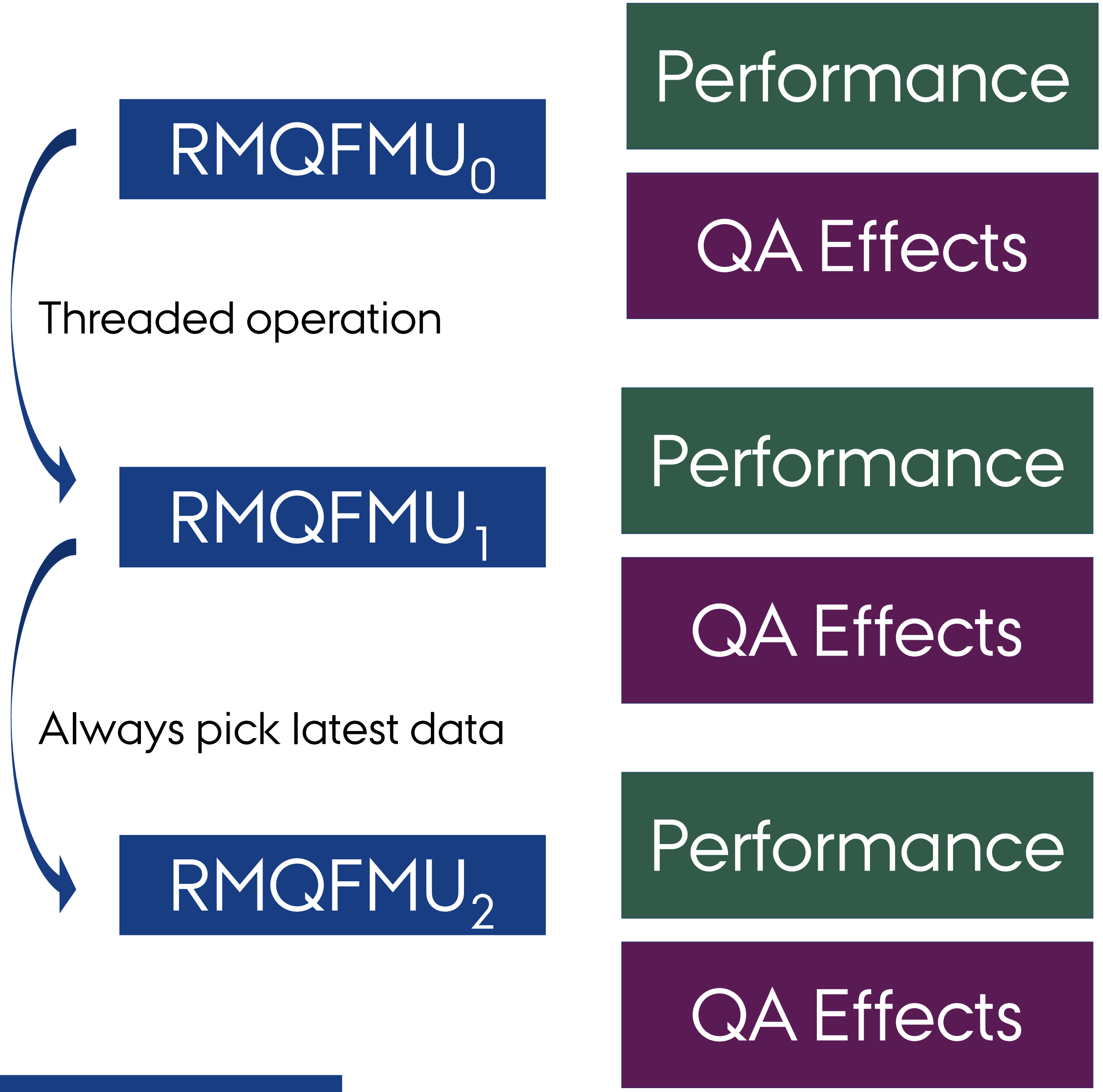
Threaded operation

RMQFMU$_1$

Always pick latest data

RMQFMU$_2$

Evaluation



RMQFMU$_0$

Performance

QA Effects

Threaded operation

RMQFMU$_1$

Performance

QA Effects

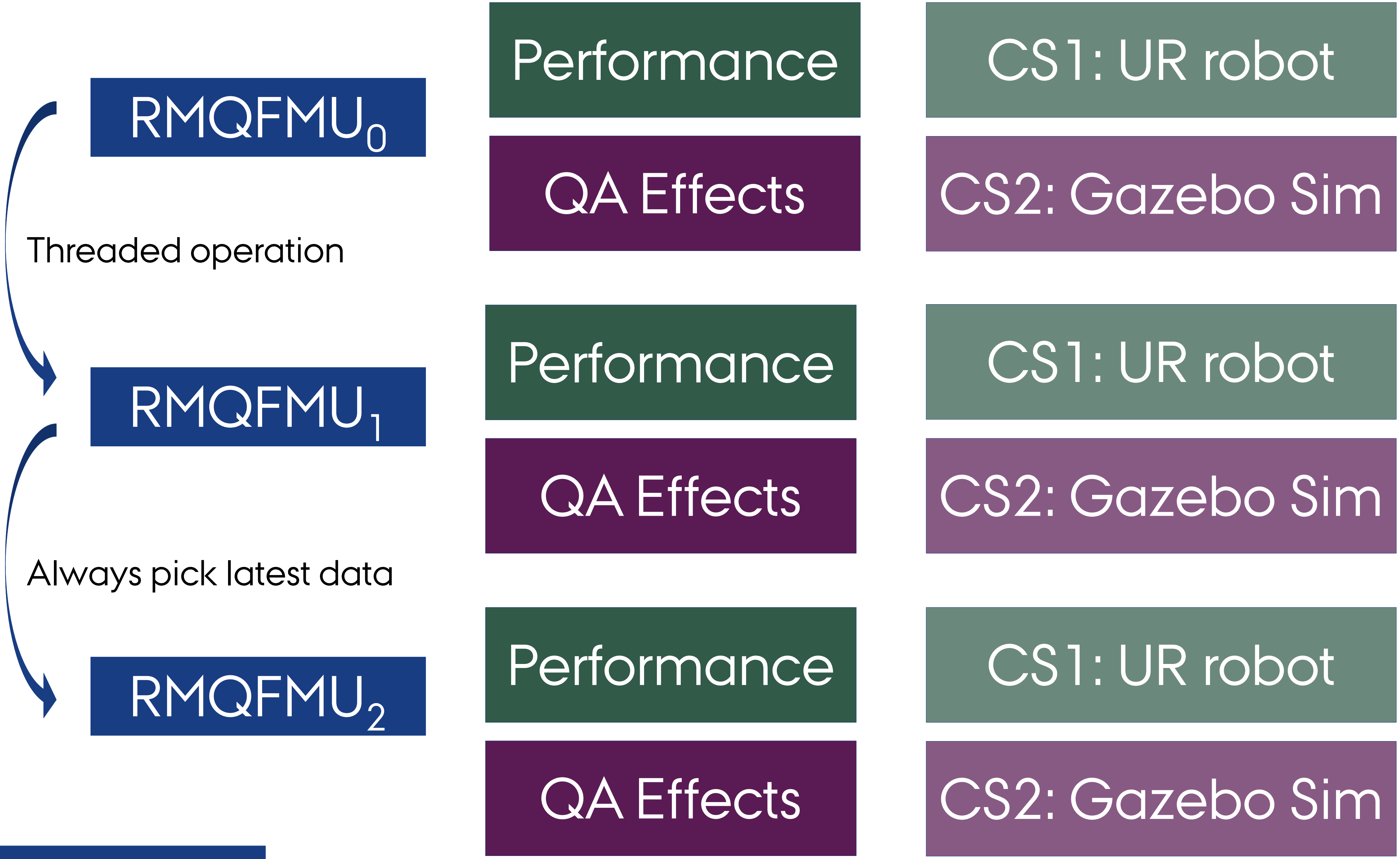
Always pick latest data

RMQFMU$_2$

Performance

QA Effects

Evaluation



RMQFMU_0

Threaded operation

RMQFMU_1

Always pick latest data

RMQFMU_2

Performance

QA Effects

CS1: UR robot

CS2: Gazebo Sim

Performance

QA Effects

CS1: UR robot

CS2: Gazebo Sim

Performance

QA Effects

CS1: UR robot

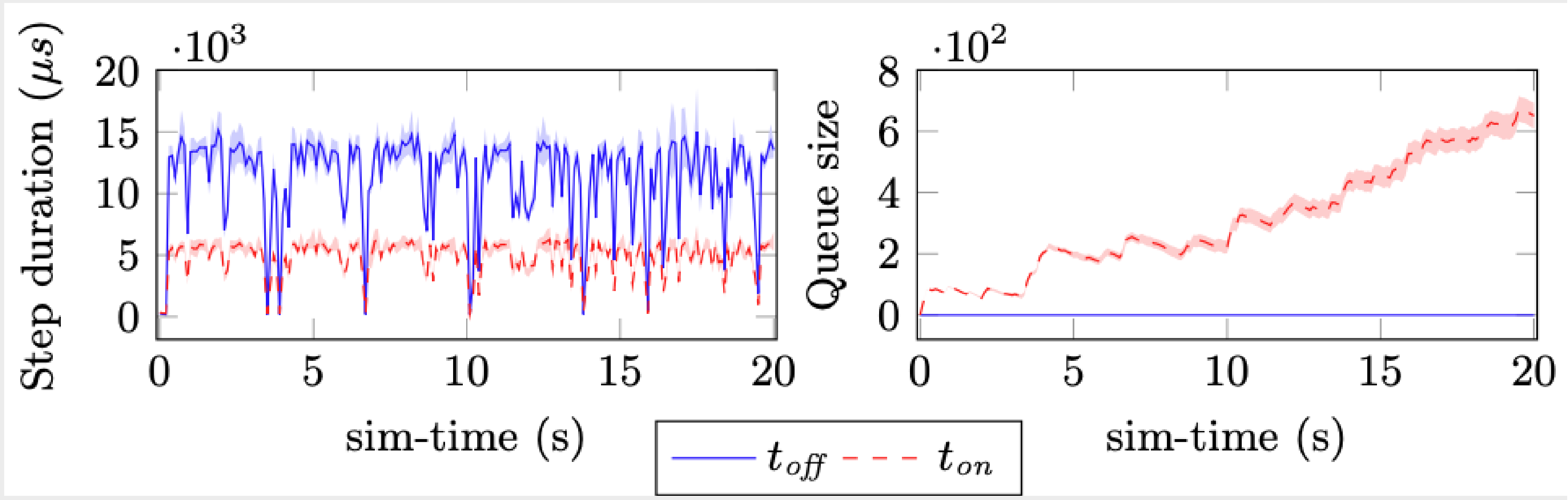
CS2: Gazebo Sim

Evaluation

Step-size = 100ms
Delay = 100ms

RMQFMU$_0$ - t_{off}

RMQFMU$_1$ - t_{on}

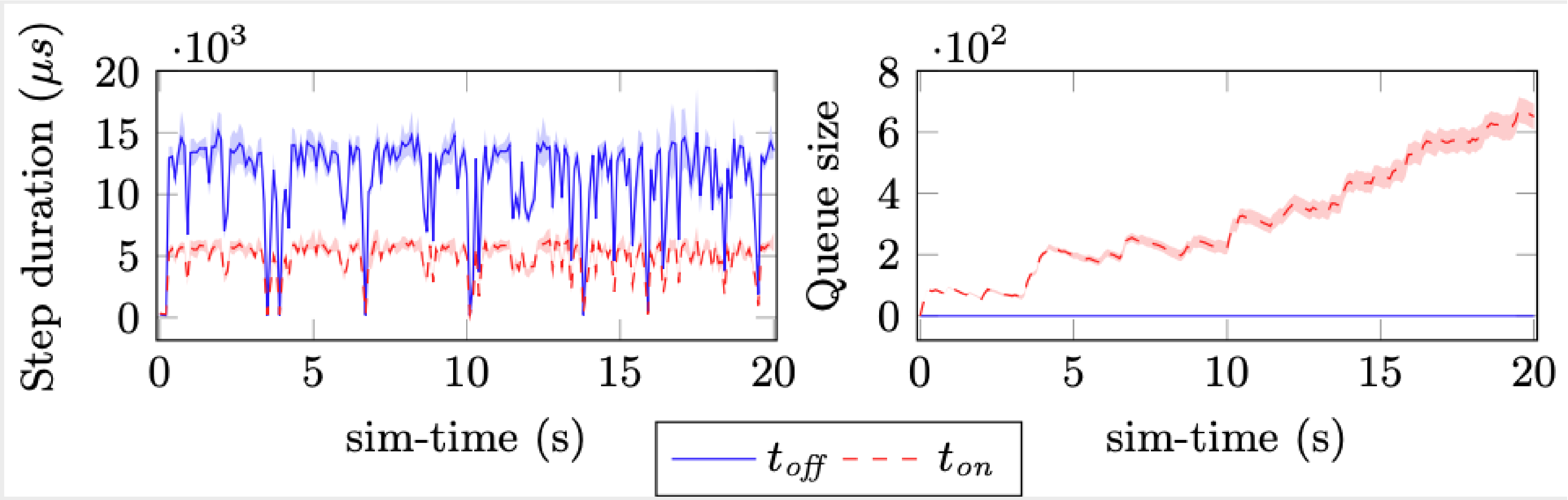


Evaluation Performance

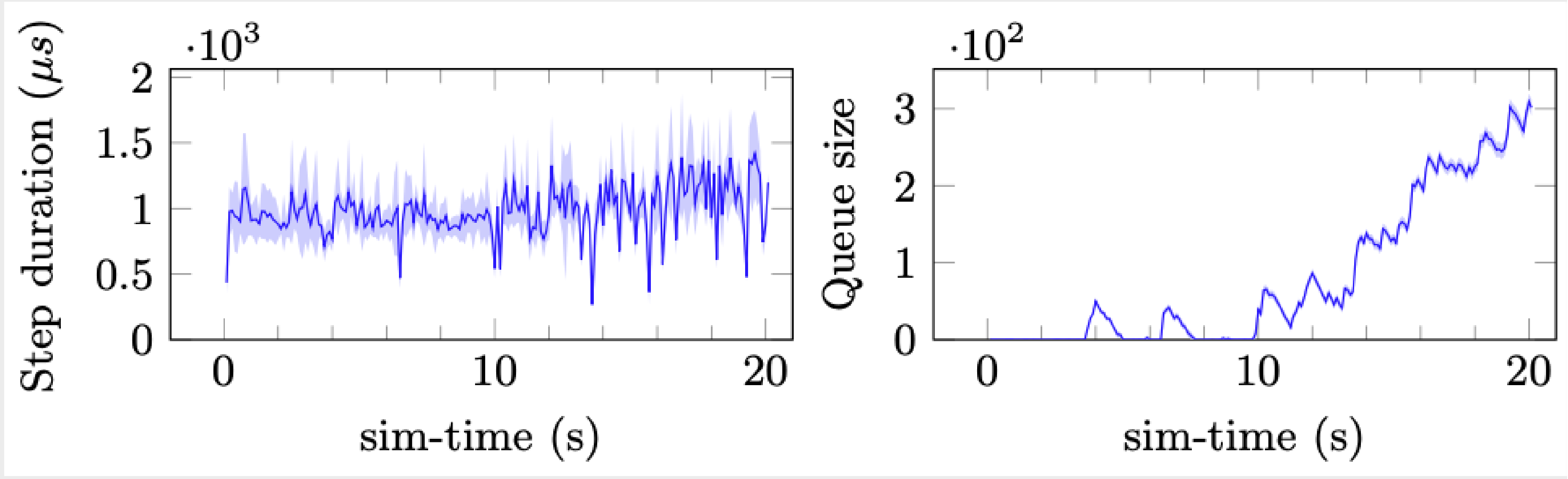
Step-size = 100ms
Delay = 100ms

RMQFMU$_0$ - t$_{off}$

RMQFMU$_1$ - t$_{on}$



RMQFMU$_2$ - t$_{on}$

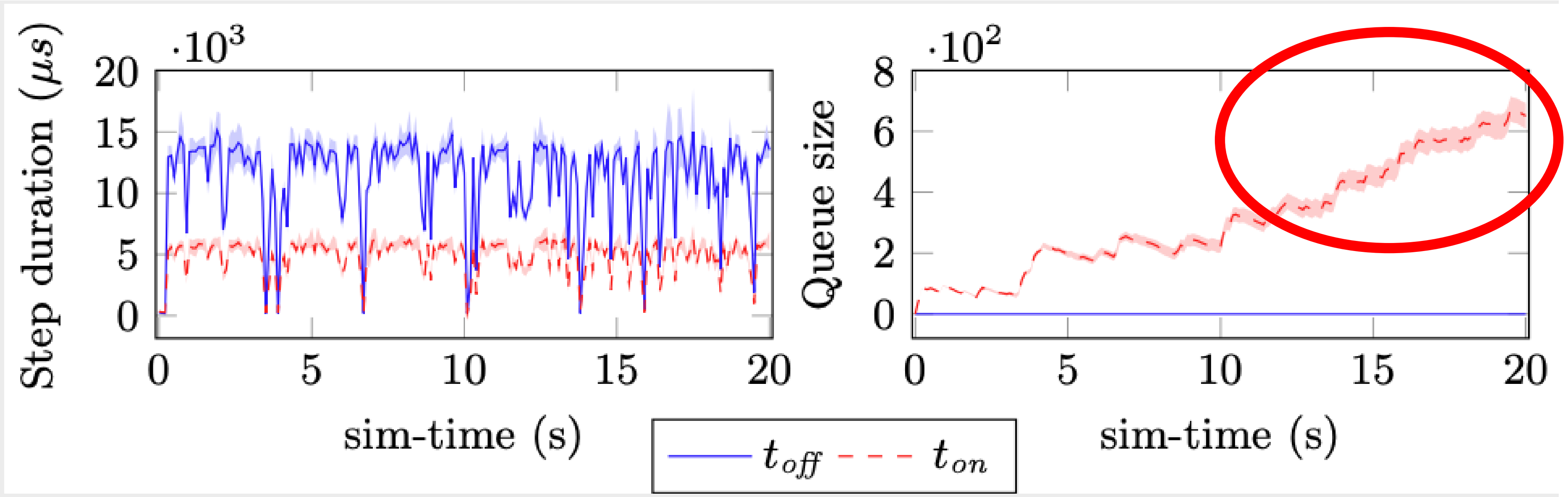


Evaluation Performance

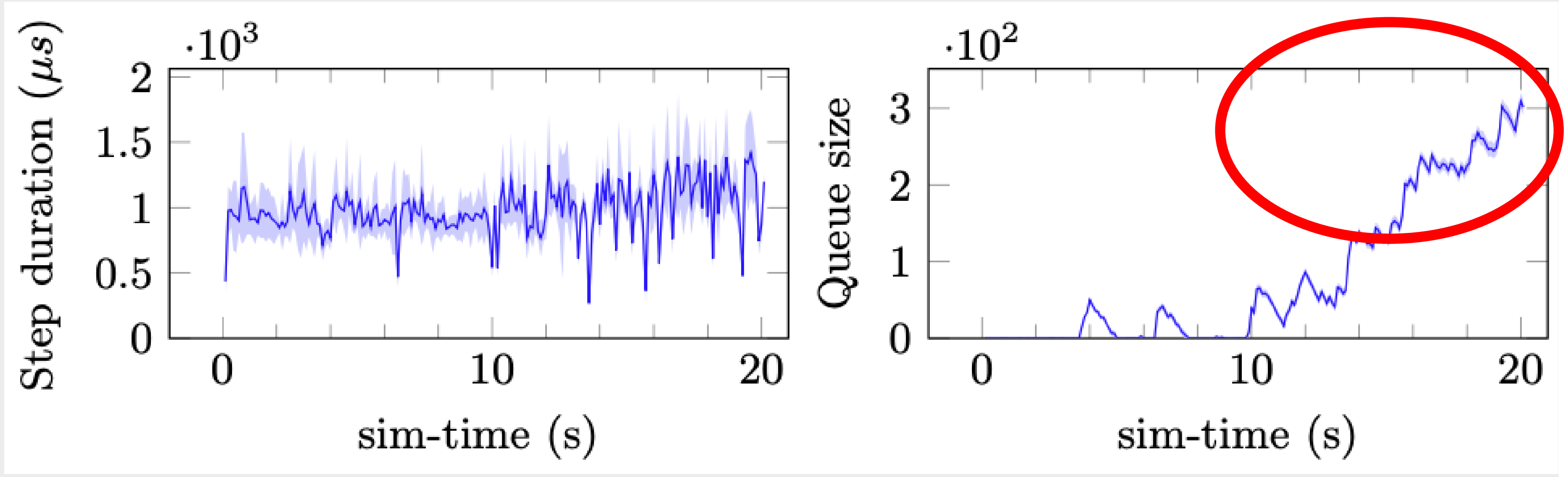
Step-size = 100ms
Delay = 100ms

RMQFMU$_0$ - t$_{off}$

RMQFMU$_1$ - t$_{on}$



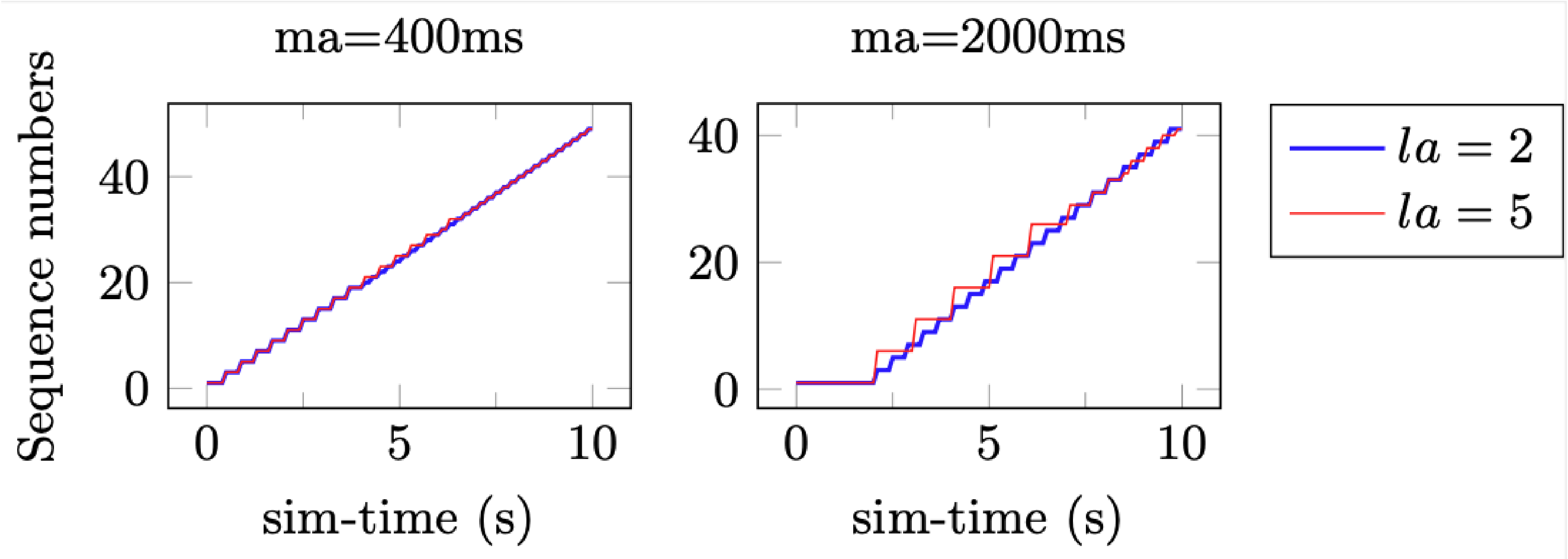
RMQFMU$_2$ - t$_{on}$



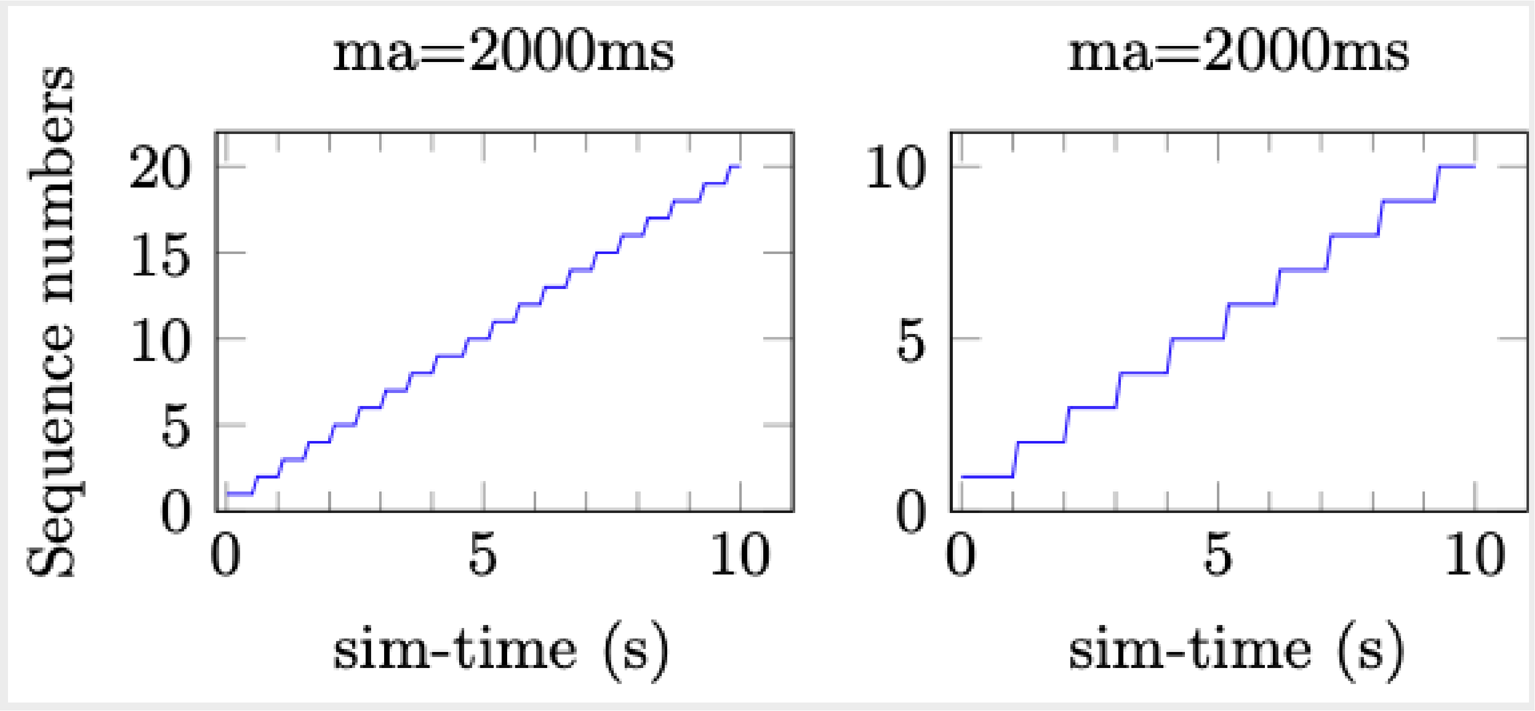
Evaluation Performance

f_{data} = 200ms

RMQFMU$_1$ - t_{on}



RMQFMU$_2$ - t$_{on}$

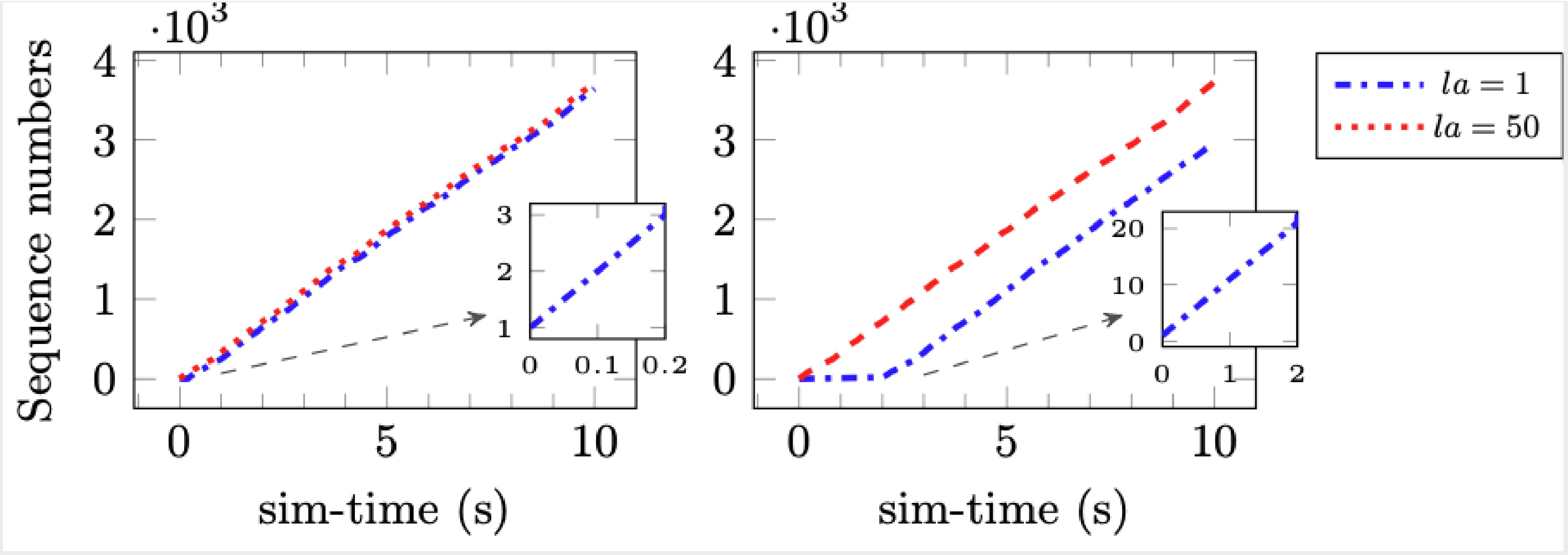


500ms gaps

1000ms gaps

f_{data} = 2ms

RMQFMU$_2$ - t_{on}



ma = 200ms

ma = 2000ms

Evaluation QA Effects

Take away

Take away

Maxage: dealing with gaps in data

Take away

Maxage: dealing with gaps in data

Lookahead: jump ahead in the queue

Take away

Maxage: dealing with gaps in data

Lookahead: jump ahead in the queue

Frequency of sending affects the lookahead

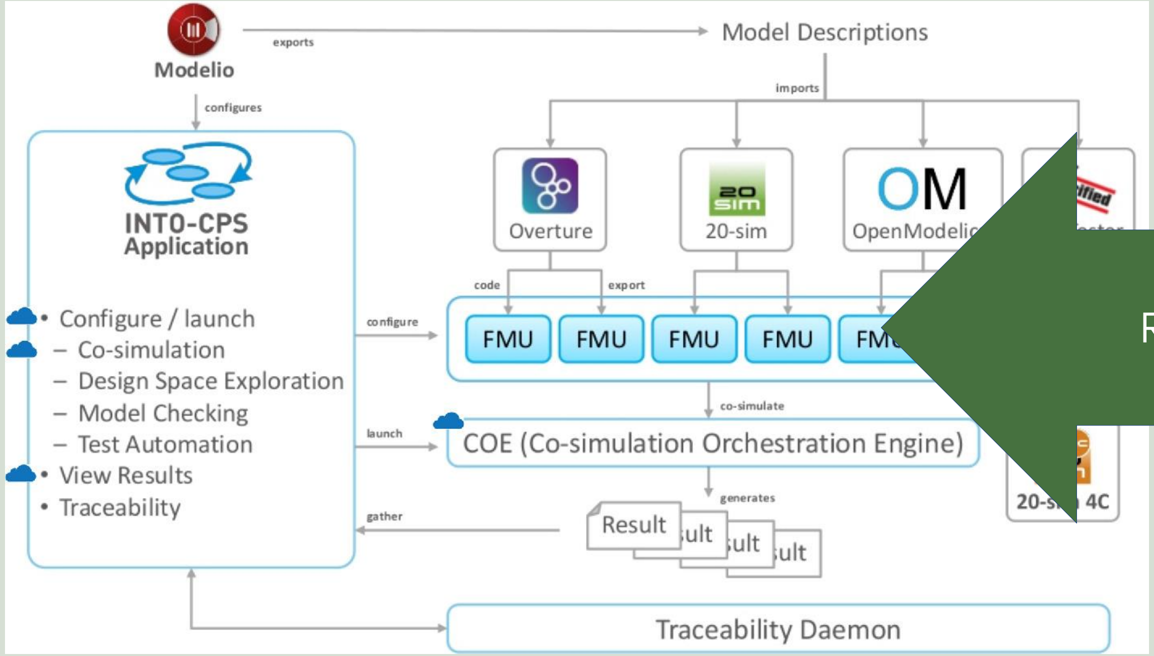
Wrapping it up

- RMQFMU
    - Why is it useful?

Wrapping it up

Co-Simulation World

'Real' World



- Configure / launch
  - Co-simulation
  - Design Space Exploration
  - Model Checking
  - Test Automation
- View Results
- Traceability

COE (Co-simulation Orchestration Engine)

Traceability Daemon

RMQFMU

A Robot

A Gazebo Simulation

Useful links

- Technical report: https://arxiv.org/pdf/2107.01010.pdf
- Latest release v2.1.2: https://github.com/INTO-CPS-Association/fmu-rabbitmq

- What does the future hold?

Wrapping it up

- What does the future hold?
  - Enabling variable step-size

Wrapping it up

- What does the future hold?
  - Enabling variable step-size
  - Formal verification of the FMU

- What does the future hold?
  - Enabling variable step-size
  - Formal verification of the FMU
  - General solution: rabbitmq, zeromq etc.

Wrapping it up

- What does the future hold?
  - Enabling variable step-size
  - Formal verification of the FMU
  - General solution: rabbitmq, zeromq etc.
  - Comparison to DB query

Wrapping it up

mirgita.frasheri@ece.au.dk

AARHUS
UNIVERSITET