

# News and Status about the FMI Standard and its Implementation in OMSimulator

(Lennart Ochel (VTI) and Arunkumar Palanisamy (Linköping University))

# FMI News and Layered Standards

# News

- 2025-10-27 Proceedings of Modelica and FMI Conference available (<https://ecp.ep.liu.se/index.php/modelica/issue/view/105>)
- 2025-09-11 FMI Advisory Committee Meeting in Lucerne
- 2025-09-08 Modelica and FMI Conference
- 2025-07-21 Release of FMI-LS-BUS v1.0
- 2025-06-02 250 Tools supporting FMI listed on the FMI tools page!
- 2025-06-02 FMI Design Meeting in Berlin

<https://fmi-standard.org/news/>

# Layered Standards

- FMI-LS-XCP          FMI Layered Standard for XCP
- FMI-LS-BUS         FMI Layered Standard for Network Communication
- FMI-LS-STRUCT    FMI Layered Standard for Structured Data
- FMI-LS-REF         FMI Layered Standard References

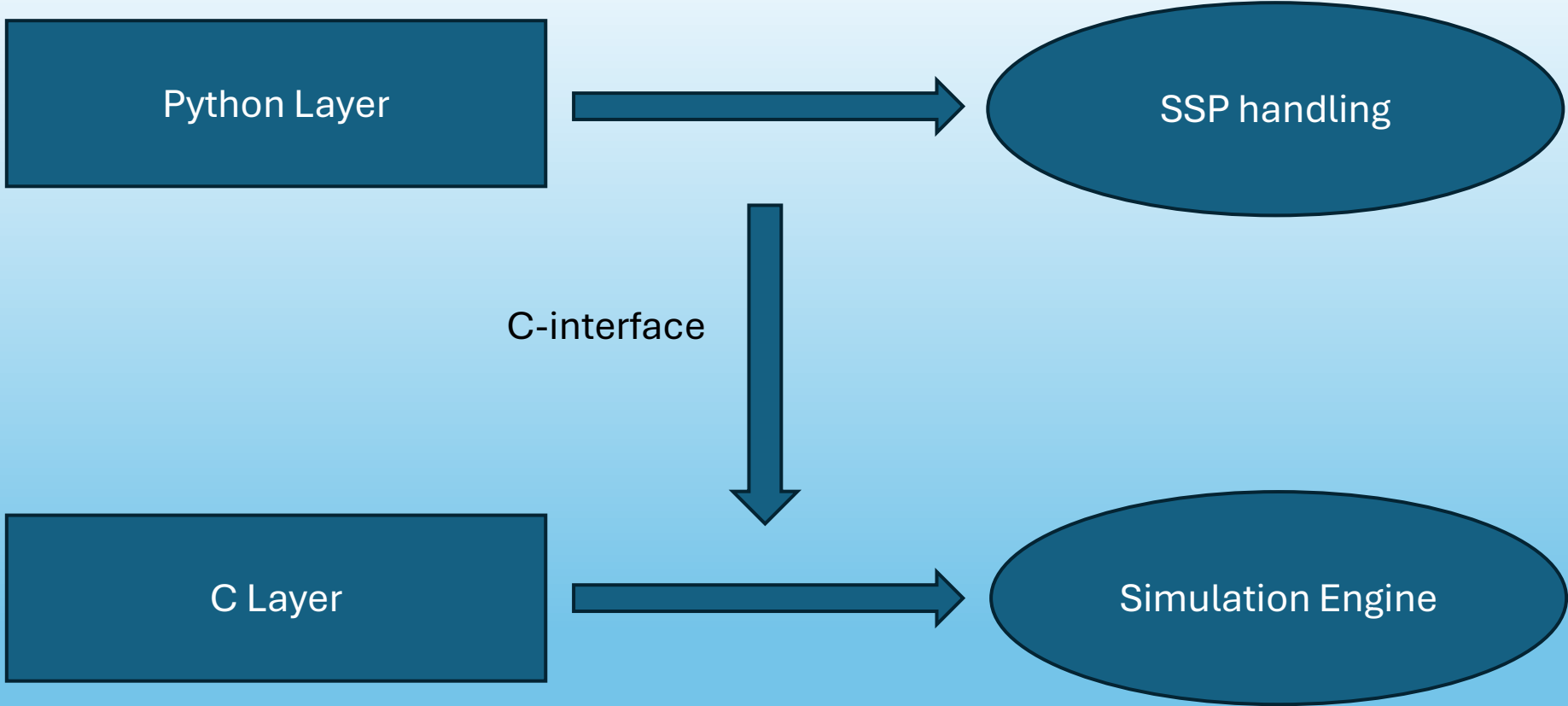
<https://github.com/modelica/fmi-standard>

# OMSimulator Updates

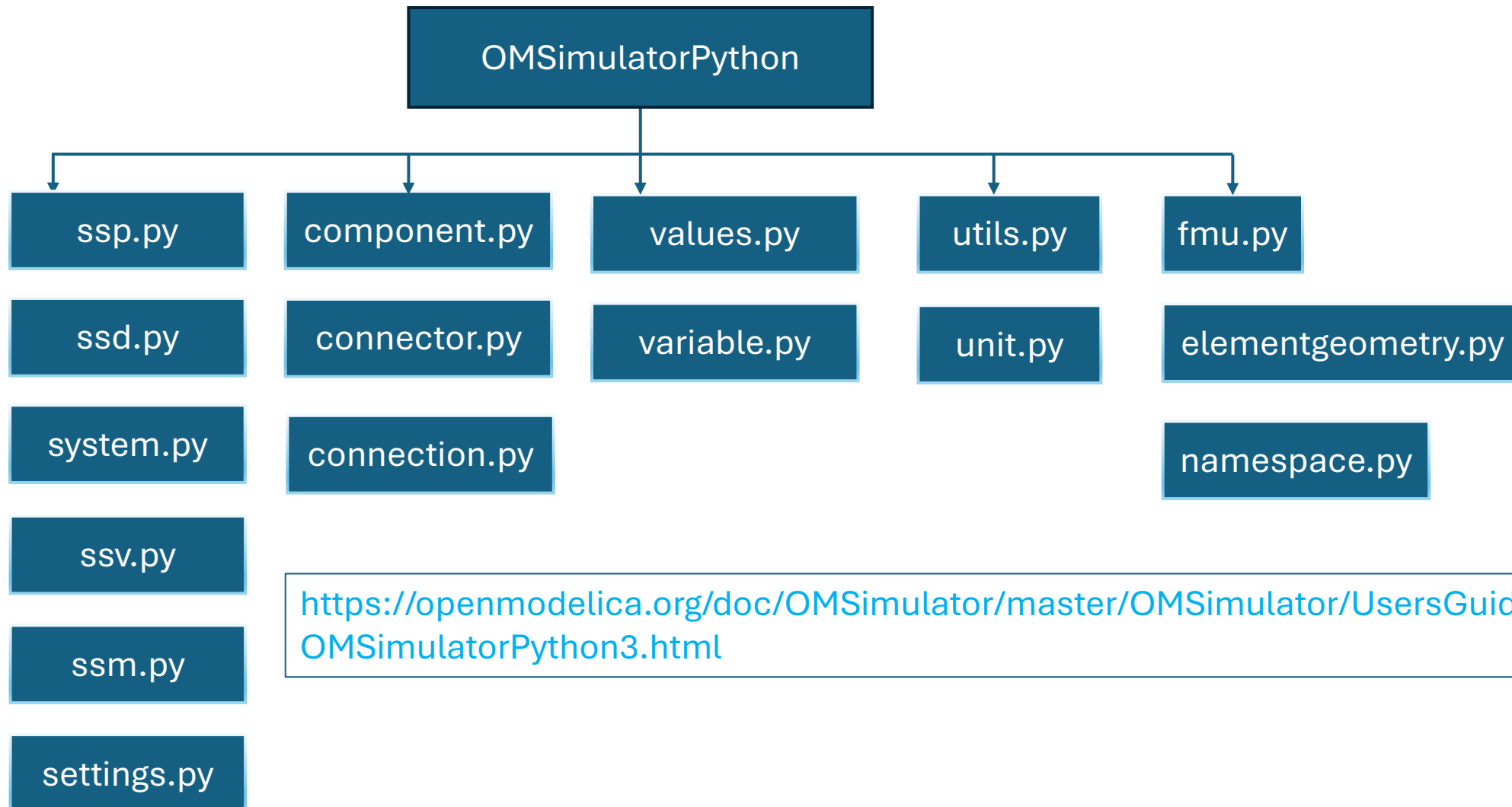
# Why split the C-API? Motivation

- Improve modularity and reduce overhead
- Enhance python integration
- Seperate simulation core from SSP standards
- Import SSP from other tool vendors
- SSP concepts are more (SSV, SSM, SSB, SRMD etc...)

# Architecture Design



# OMSimulator Python Package



# Example 1: Simple FMU simulation

```
from OMSimulator import FMU

fmu = FMU('Modelica.Blocks.Math.Gain.fmu')

print('name', fmu.modelName)
print('guid', fmu.guid)
print('fmi version', fmu.fmiVersion)

print('States:')
for var in sorted(fmu.states, key=lambda x: x.name):
    print({'name': var.name, 'signal_type': var.signal_type.name,
          'valueReference': var.valueReference, 'variability': var.variability,
          'causality': var.causality.name})

fmu.instantiate()
fmu.setResultFile('Gain_res.mat')
fmu.setValue('k', 3.0)

fmu.initialize()
fmu.simulate()
fmu.terminate()

fmu.delete()
```

# Example 2: SSP import/export and simulation

```
from OMSimulator import SSP, Settings, CRef

model = SSP()
model.addResource('../resources/Modelica.Blocks.Math.Add.fmu', new_name='resources/Add.fmu')
model.addResource('../resources/Modelica.Blocks.Math.Gain.fmu', new_name='resources/Gain.fmu')

model.addComponent(CRef('default', 'Add1'), 'resources/Add.fmu')
model.addComponent(CRef('default', 'Gain1'), 'resources/Gain.fmu')

model.addConnection(CRef('default', 'Gain1', 'y'), CRef('default', 'Add1', 'u1'))

model.export('SimpleSimulation2.ssp')

## Import model again and simulate
model2 = SSP('SimpleSimulation2.ssp')
instantiated_model = model2.instantiate()
instantiated_model.setResultFile("SimpleSimulation2_res.mat")
instantiated_model.initialize()
instantiated_model.simulate()
```

# Example 3: Architectural modelling

```
from OMSimulator import CRef, Settings, Connector, Causality, SignalType, SSV
from OMSimulator.component import Component

component = Component(CRef('add'), 'dummy.fmu')

## add Connectors
component.addConnector(Connector('input', Causality.input, SignalType.Real))
component.addConnector(Connector('output', Causality.output, SignalType.Real))
component.addConnector(Connector('parameter', Causality.parameter, SignalType.Real))
component.setValue('parameter', 1.0)
component.setValue('input', 2.0)

## create a new SSV file and export it
ssv = SSV()
ssv.setValue('parameter', 1.0)
ssv.setValue('input', 2.0)
ssv.export('dummy.ssv')

component.addSSVReference('dummy.ssv') # Add ssv referece to the component
```

# Example 4: Mixed FMI2 and FMI3 simulation

```
model = SSP()

## FMI2.0
model.addResource("../resources/Modelica.Blocks.Math.Gain.fmu", new_name="resources/Gain.fmu")
model.addComponent(CRef("default", "Gain1"), "resources/Gain.fmu")

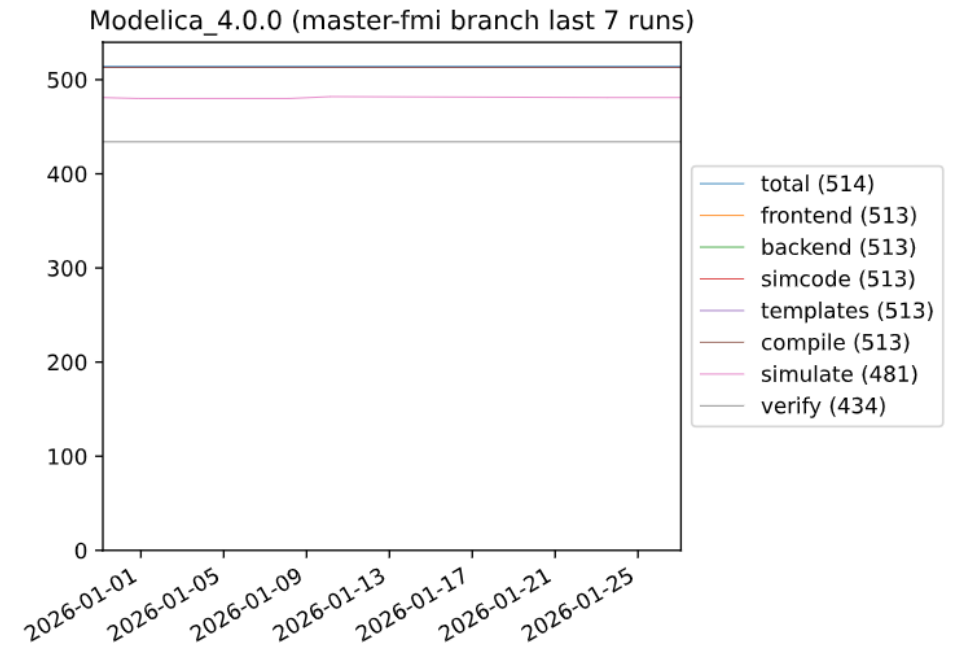
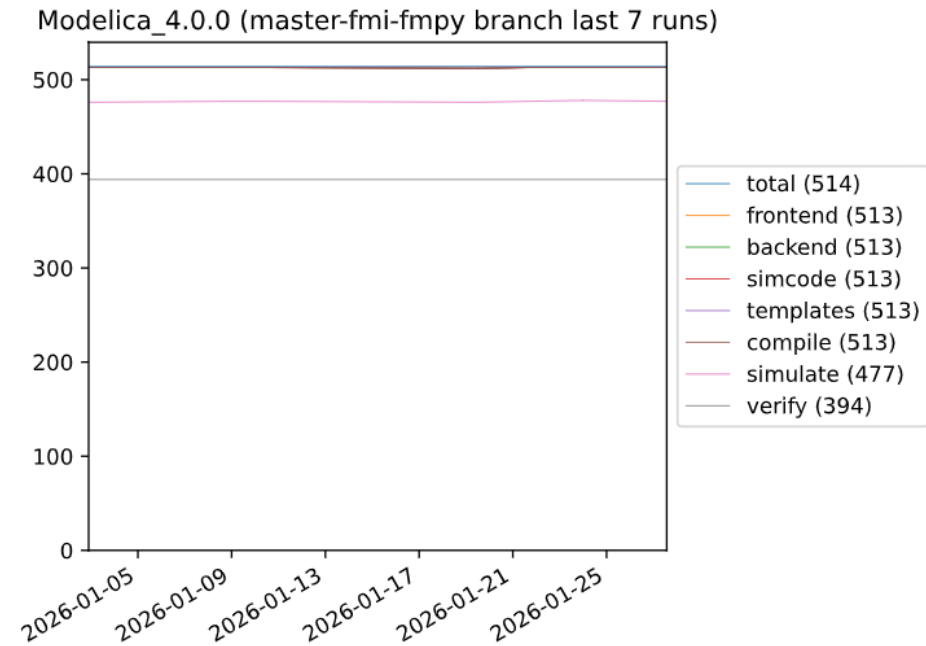
## FMI3.0
model.addResource("../resources/Feedthrough3.fmu", new_name="resources/Feedthrough3.fmu")
model.addComponent(CRef("default", "Feedthrough1"), "resources/Feedthrough3.fmu")

## connections (Gain.y (FMI2) --> Feedthrough.Float64_continuous_input (FMI3))
model.addConnection(CRef("default", "Gain1", "y"),
                    CRef("default", "Feedthrough1", "Float64_continuous_input"))

instantiated_model = model.instantiate()
instantiated_model.setResultFile("")
instantiated_model.setStopTime(10.0)
instantiated_model.setTolerance(1e-5)

instantiated_model.setValue(CRef("default", "Gain1", "u"), 15)
instantiated_model.initialize()
instantiated_model.simulate()
```

# Coverage: Fmpy vs OMSimulator



# Status

- All the API are implemented in new Python package
- All the test cases are ported to the new Python package
- Available via pip
- Full Simulation support with the new design
- Support for SSP from other tools
- FMI 3.0 CS&ME support is implemented
- Mixed-fmi support (connecting FMI-2 and FMI-3)

# TODO

- Support Missing SSP concepts
- Integrate OMS 3.0 with OMEdit