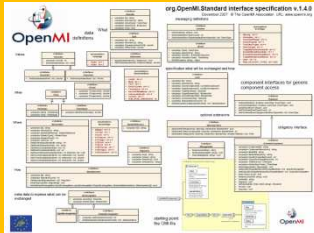


Making it easier

Jan Gregersen LicTek / DHI

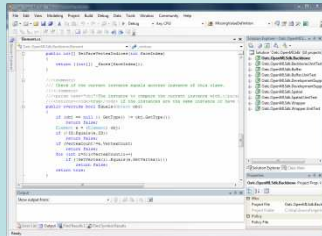
Balancing stability and flexibility



The OpenMI **Standard**

- The OpenMI.Standard interfaces
- The OpenMI standard definition
- XSD's

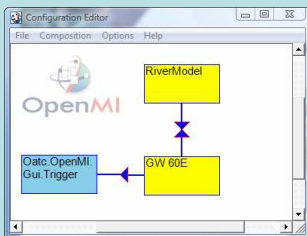
- Provided by OA
- Rigid release procedure
- Non frequent releases
- The one and only Standard
- Makes components linkable



The OpenMI **SDK**

- Source code C# (and Java)
- Default implementaion
- Wrapper
- Targeting developers

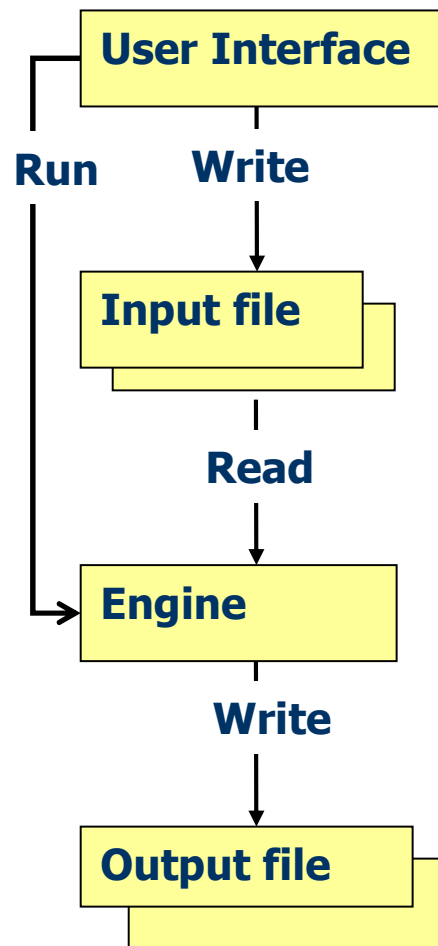
- Provided by OATC
- Flexible release procedures
- Frequent releases
- One of many
- Makes OpenMI easier
- OATC.SDK targeting models
- Not required
- Components compliant to same version of the standard can be linked regardless of which SDK is used.



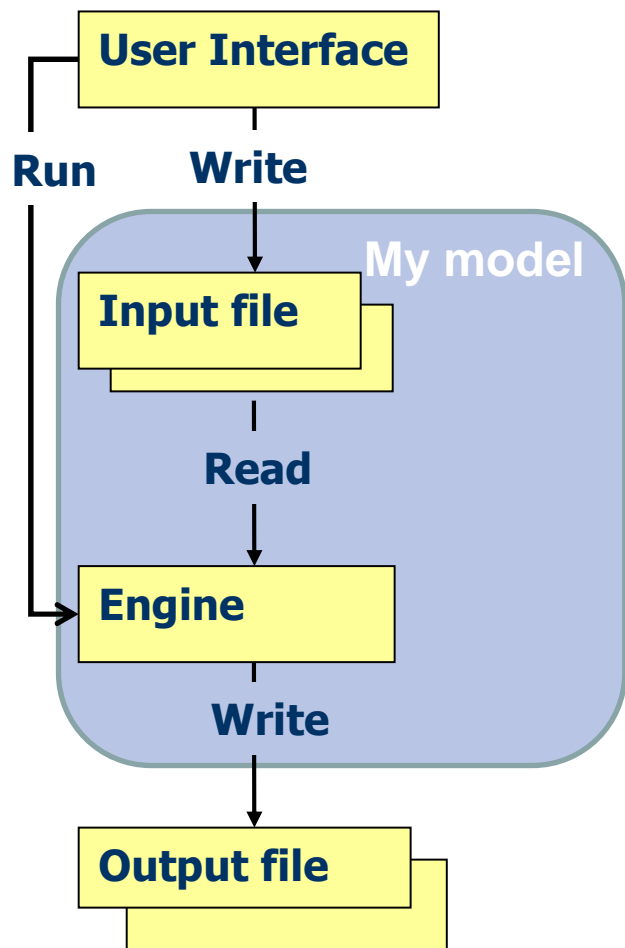
The OpenMI **Configuration Editor**

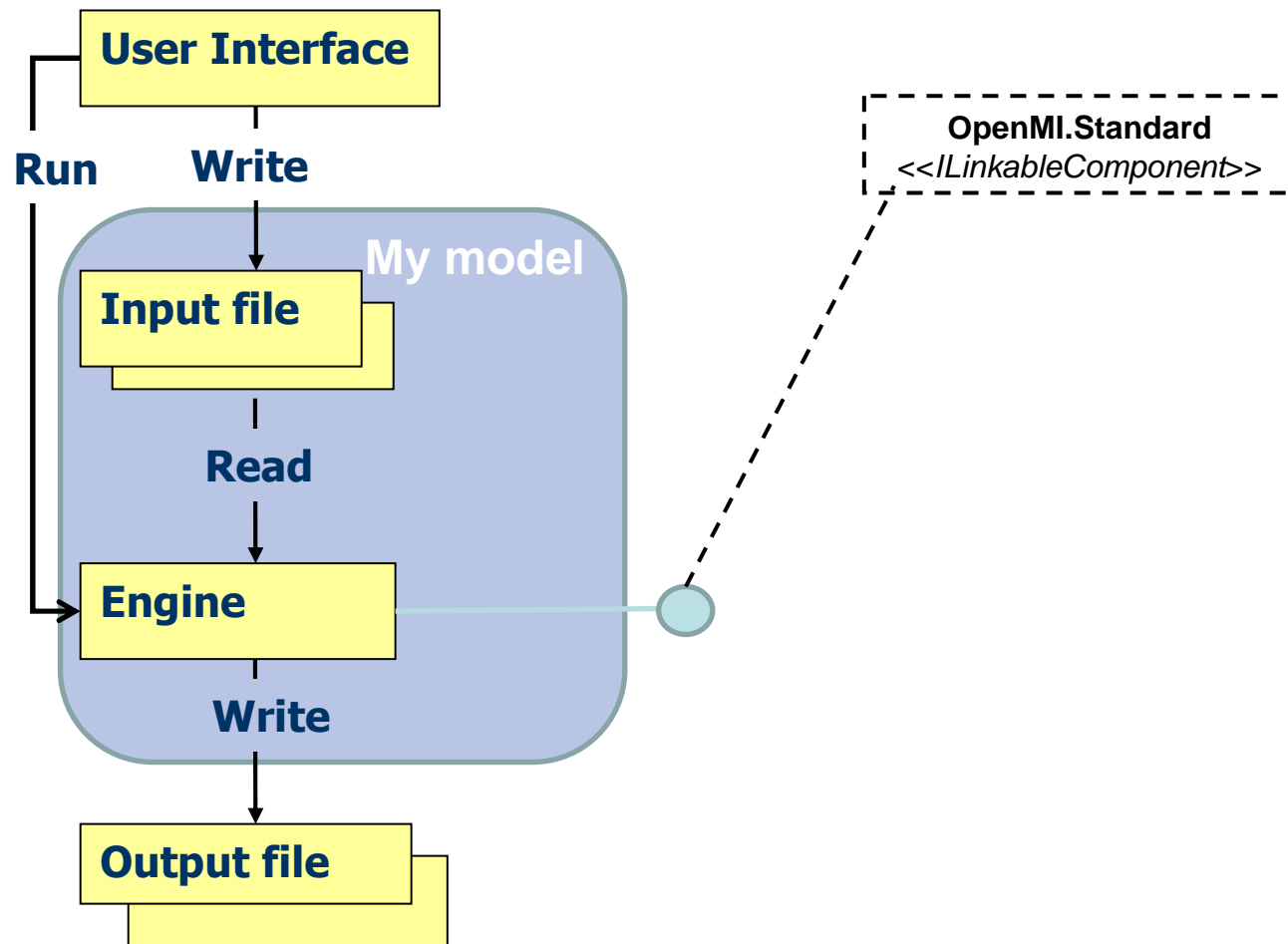
- Create linked systems
- Run linked systems
- Targeting users (e.g. modellers)

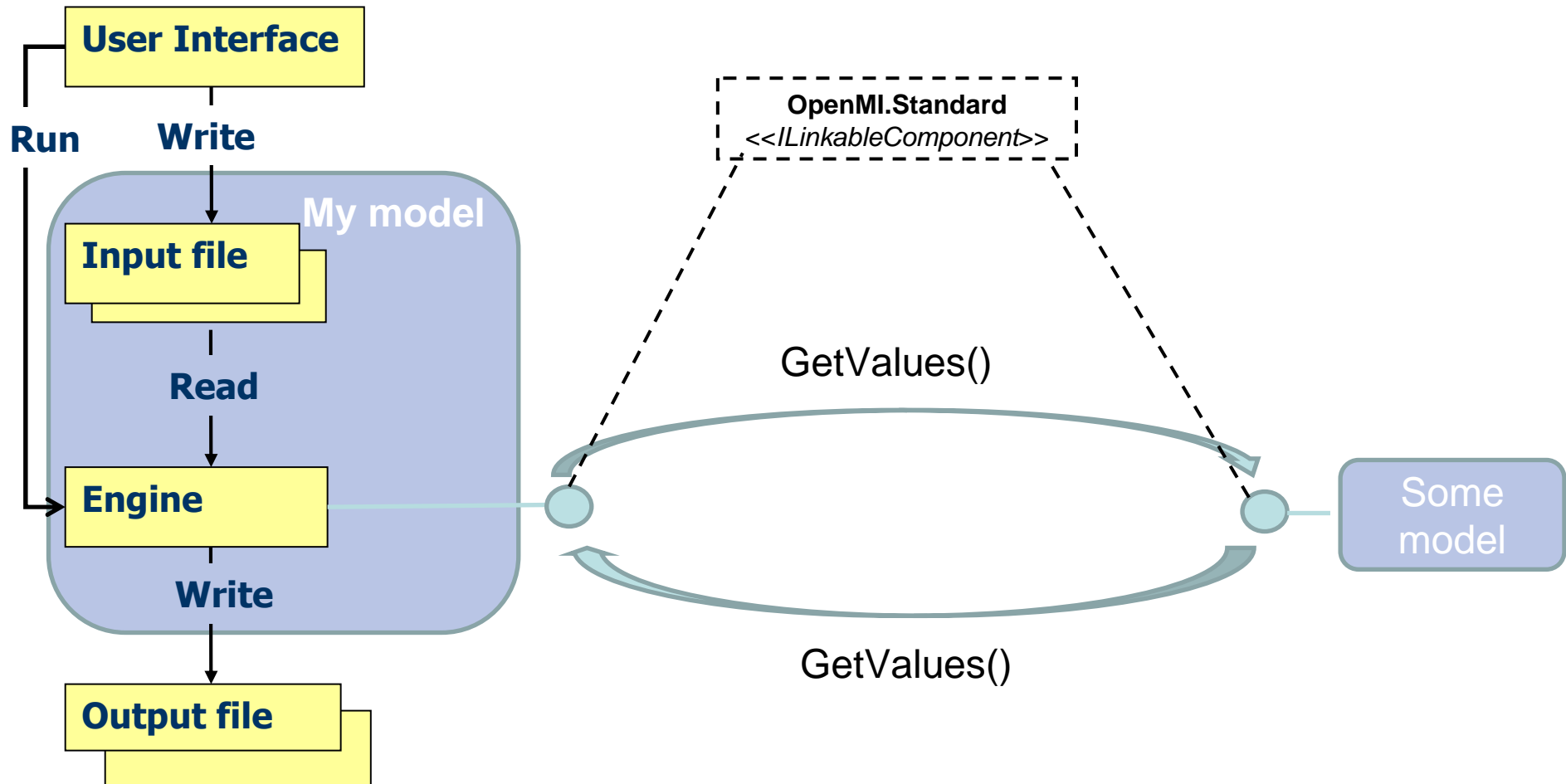
How most models are organized



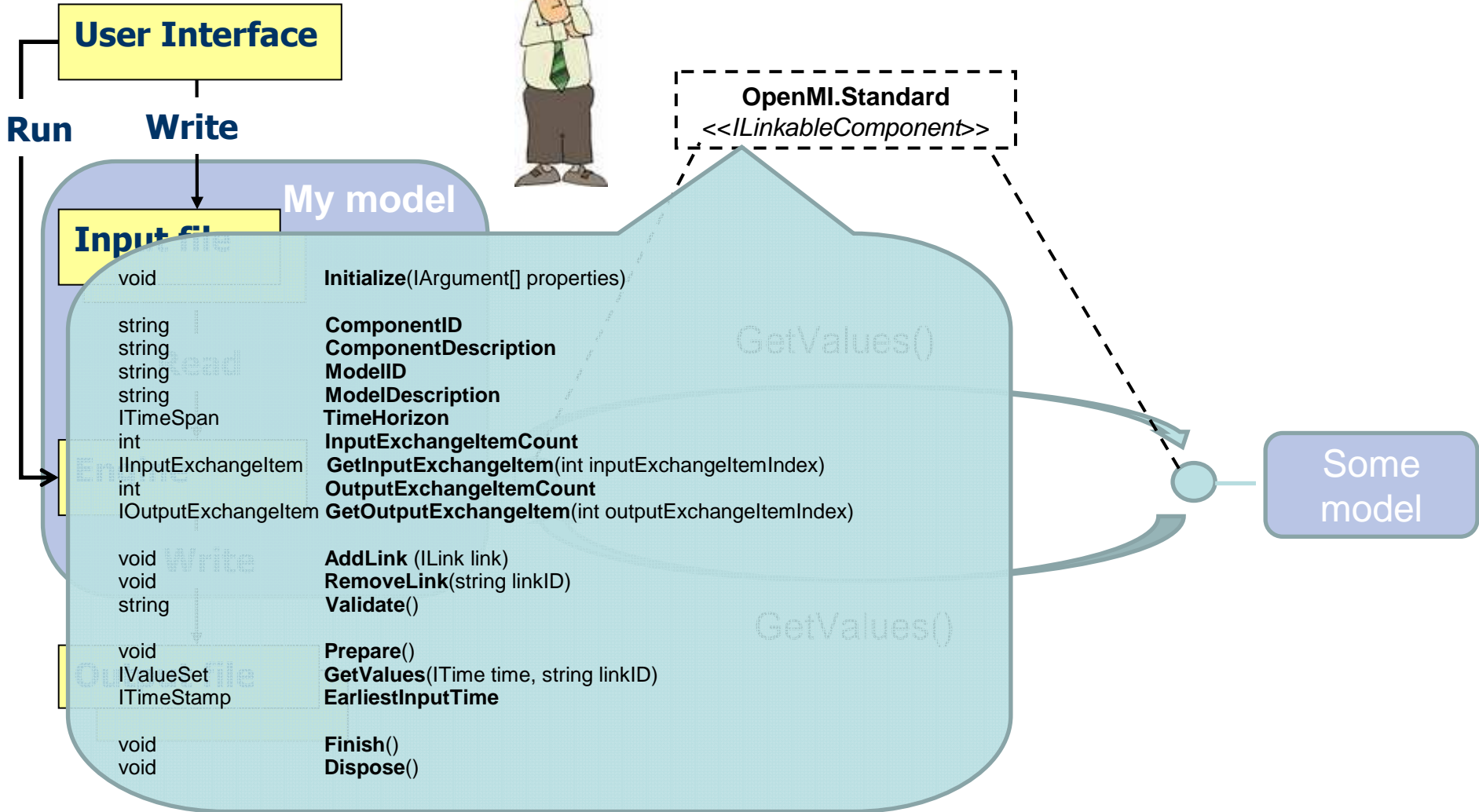
OpenMI is linking models

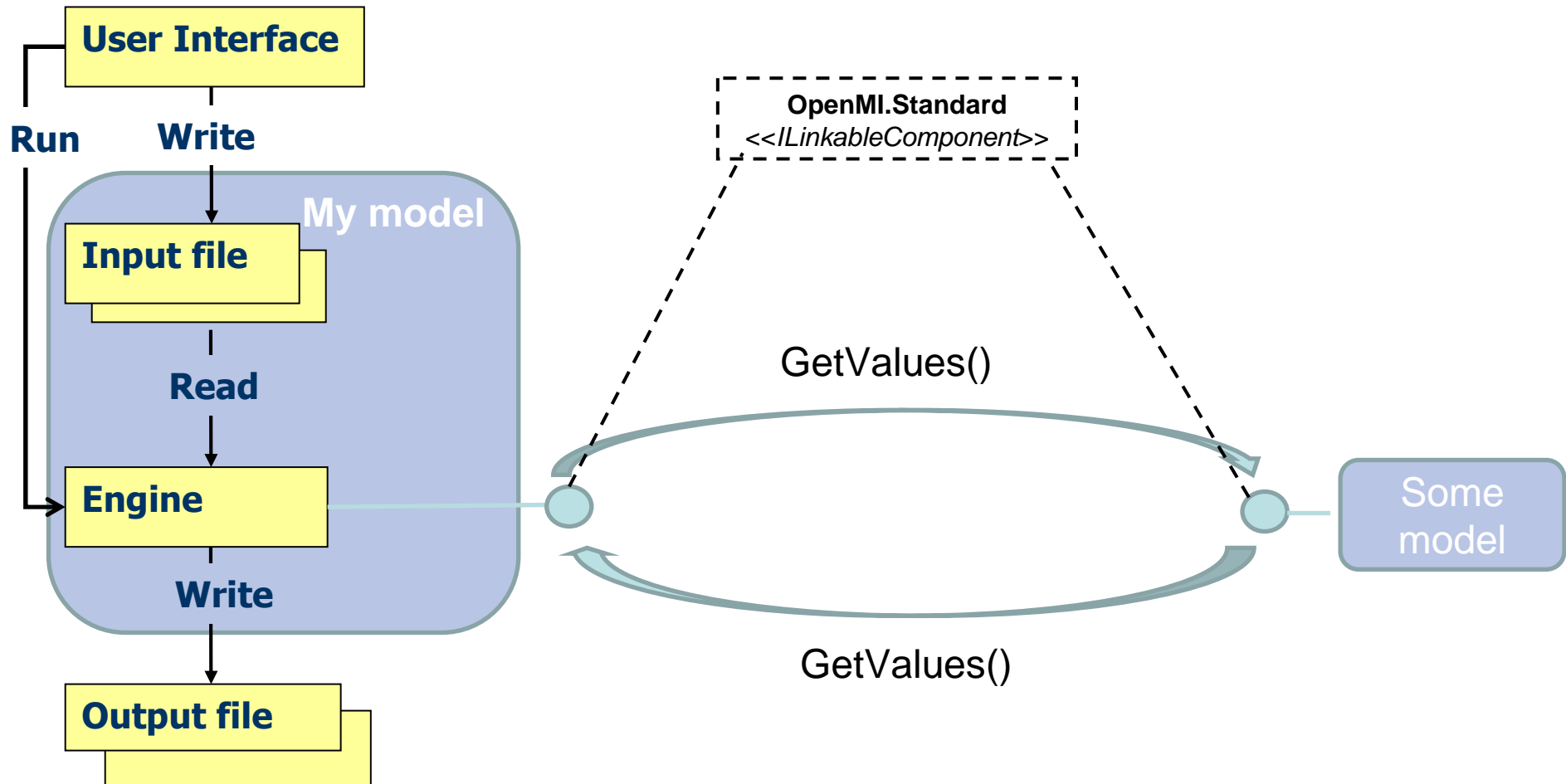


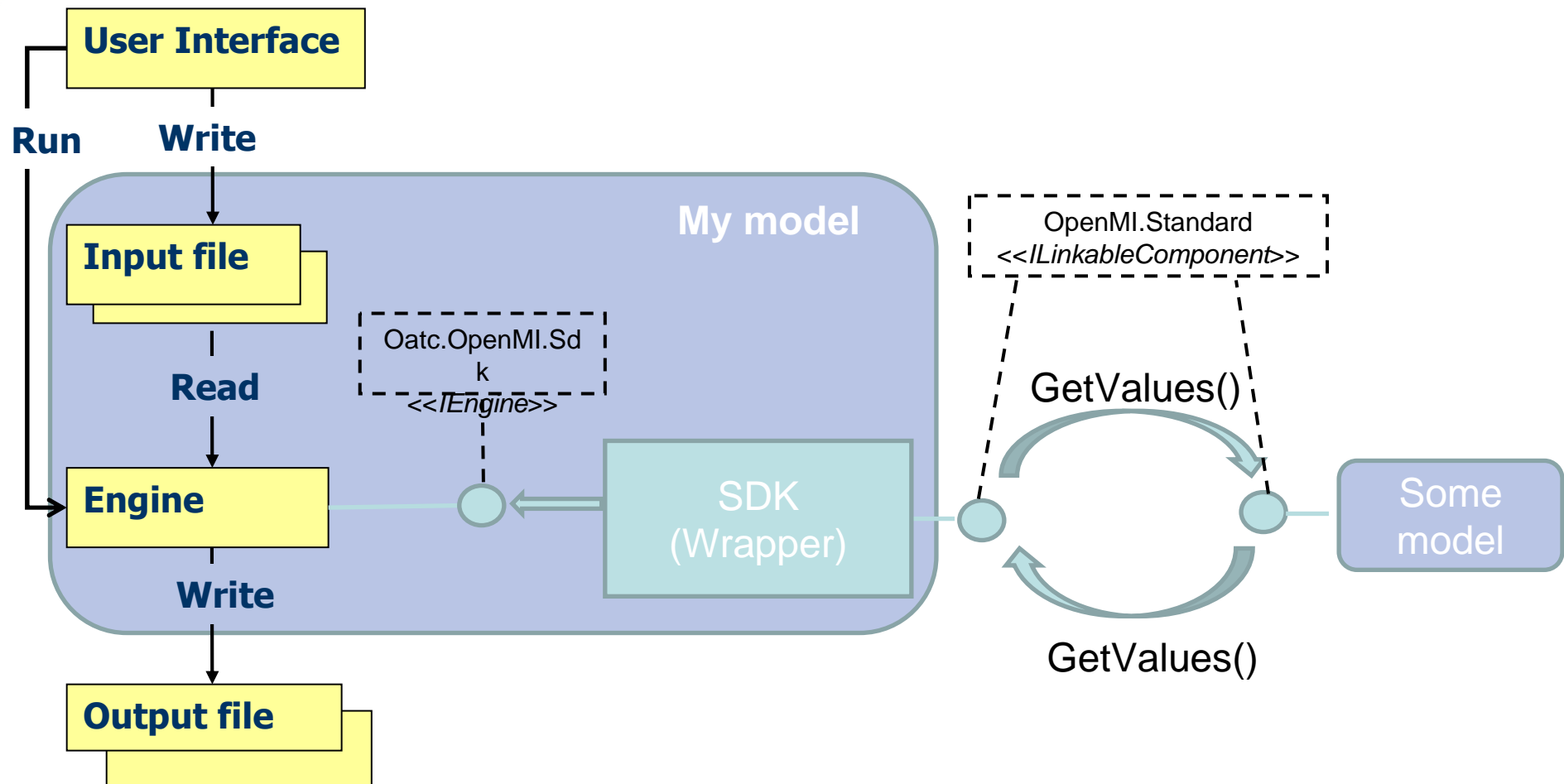


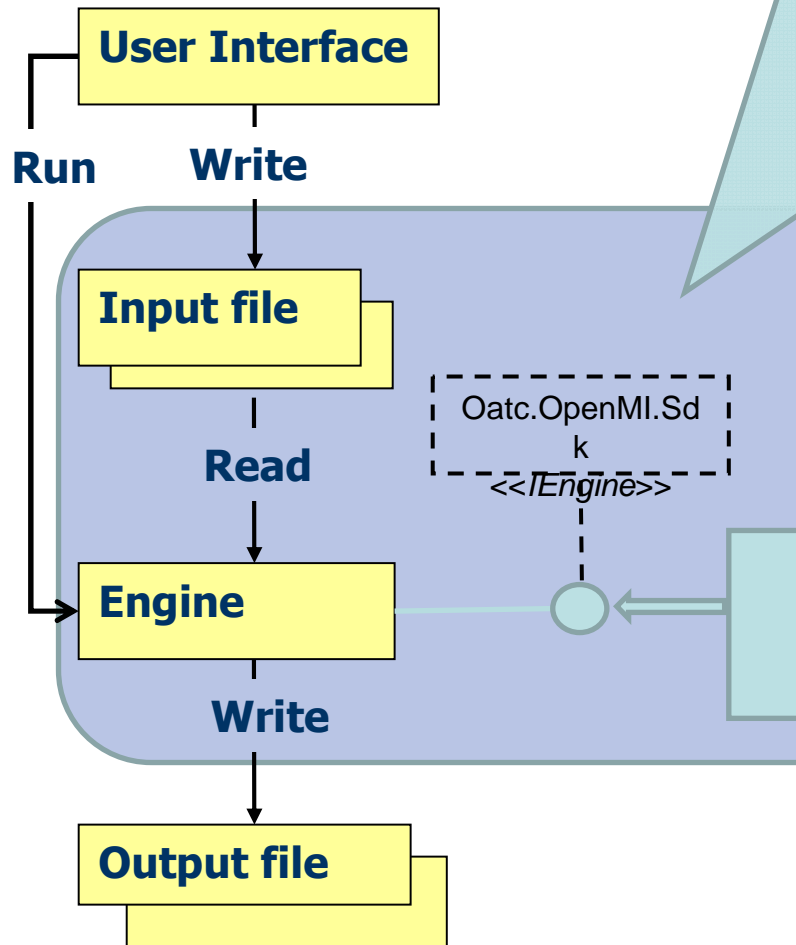
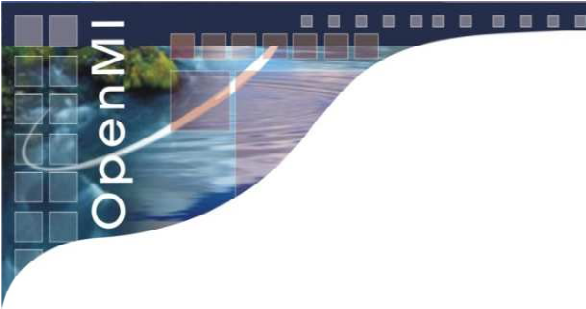


How do I set values ?









```

// -- Execution control methods (Inherited from IRunEngine) --
void Initialize(Hashtable properties);
bool PerformTimeStep();
void Finish();

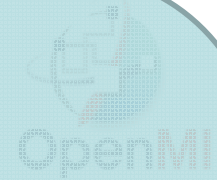
//-- Time methods (Inherited from IRunEngine) --
ITime GetCurrentTime();
ITime GetInputTime(string QuantityID, string ElementSetID);
ITimeStamp GetEarliestNeededTime();

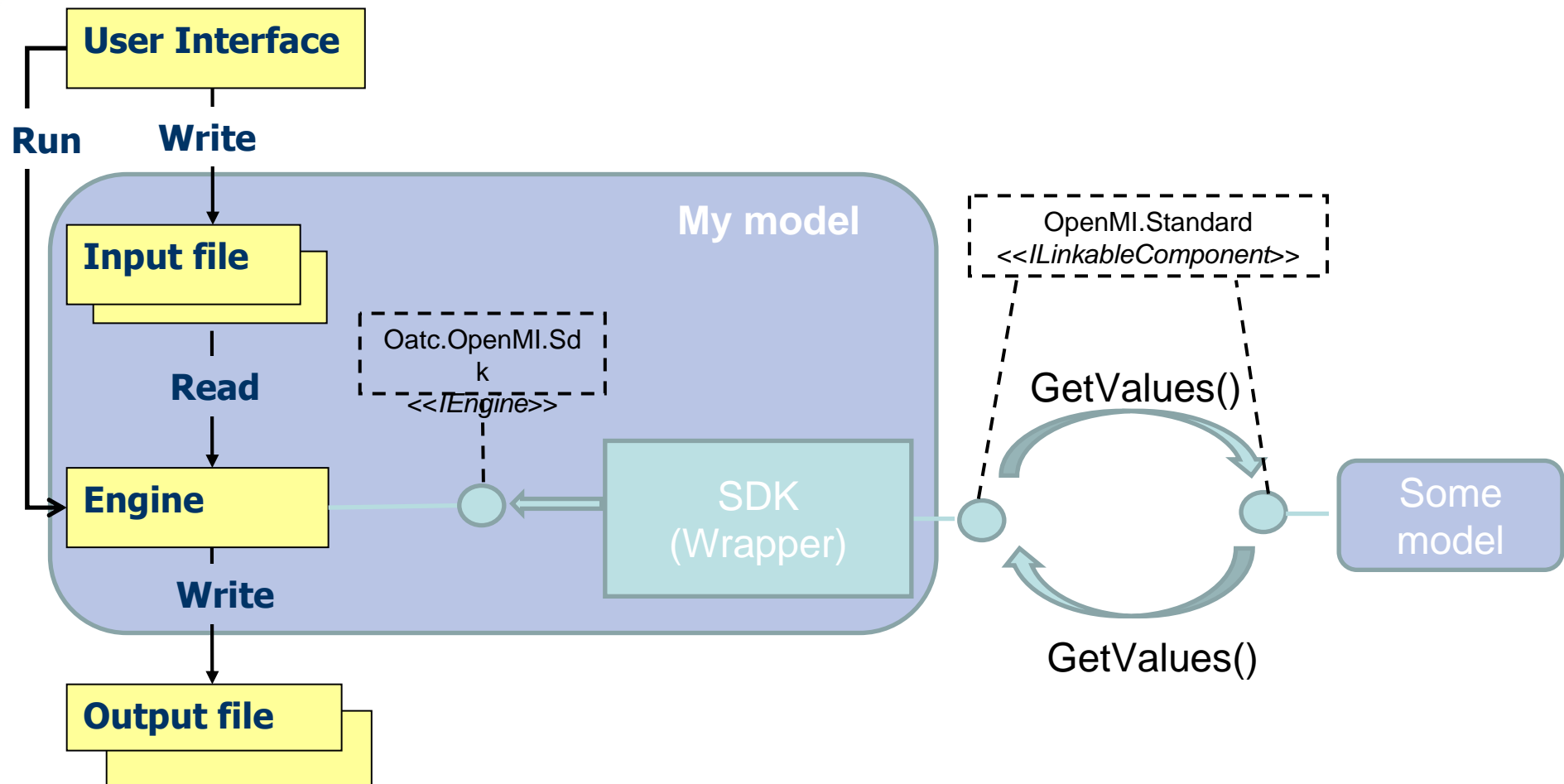
//-- Data access methods (Inherited from IRunEngine) --
void SetValues(string QuantityID, string ElementSetID, IValueSet values);
IValueSet GetValues(string QuantityID, string ElementSetID);

//-- Component description methods (Inherited from IRunEngine) --
double GetMissingValueDefinition();
string GetComponentID();
string GetComponentDescription();

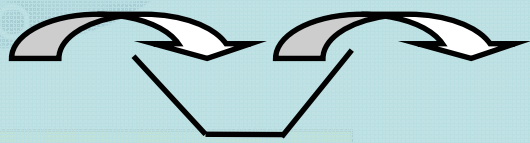
// -- Model description methods --
string GetModelID();
string GetModelDescription();
double GetTimeHorizon();

// -- Exchange items --
int GetInputExchangeItemCount();
int GetOutputExchangeItemCount();
org.OpenMI.Backbone GetInputExchangeItem(int exchangeItemIndex);
org.OpenMI.Backbone GetOutputExchangeItem(int exchangeItemIndex);
  
```

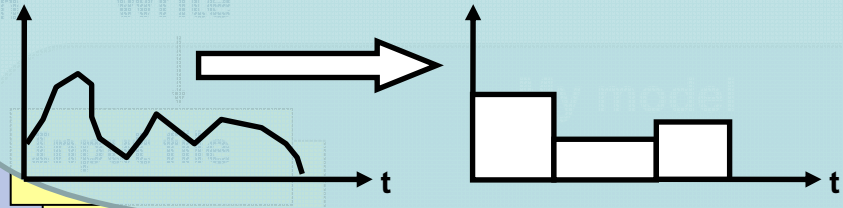




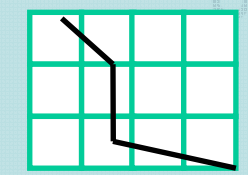
Buffers results from the engine core



Mapping of values associated to one array of times /timespans to values represented on another array of times/timespans



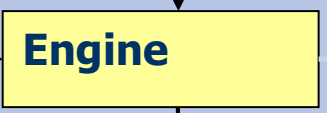
Mapping of values associated to one ElementSet to be represented on another ElementSet



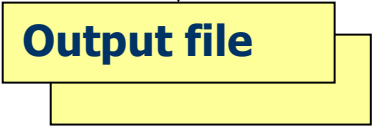
- Link administration
- Exception handling
- Events handling
- And more..

R

Read



Write



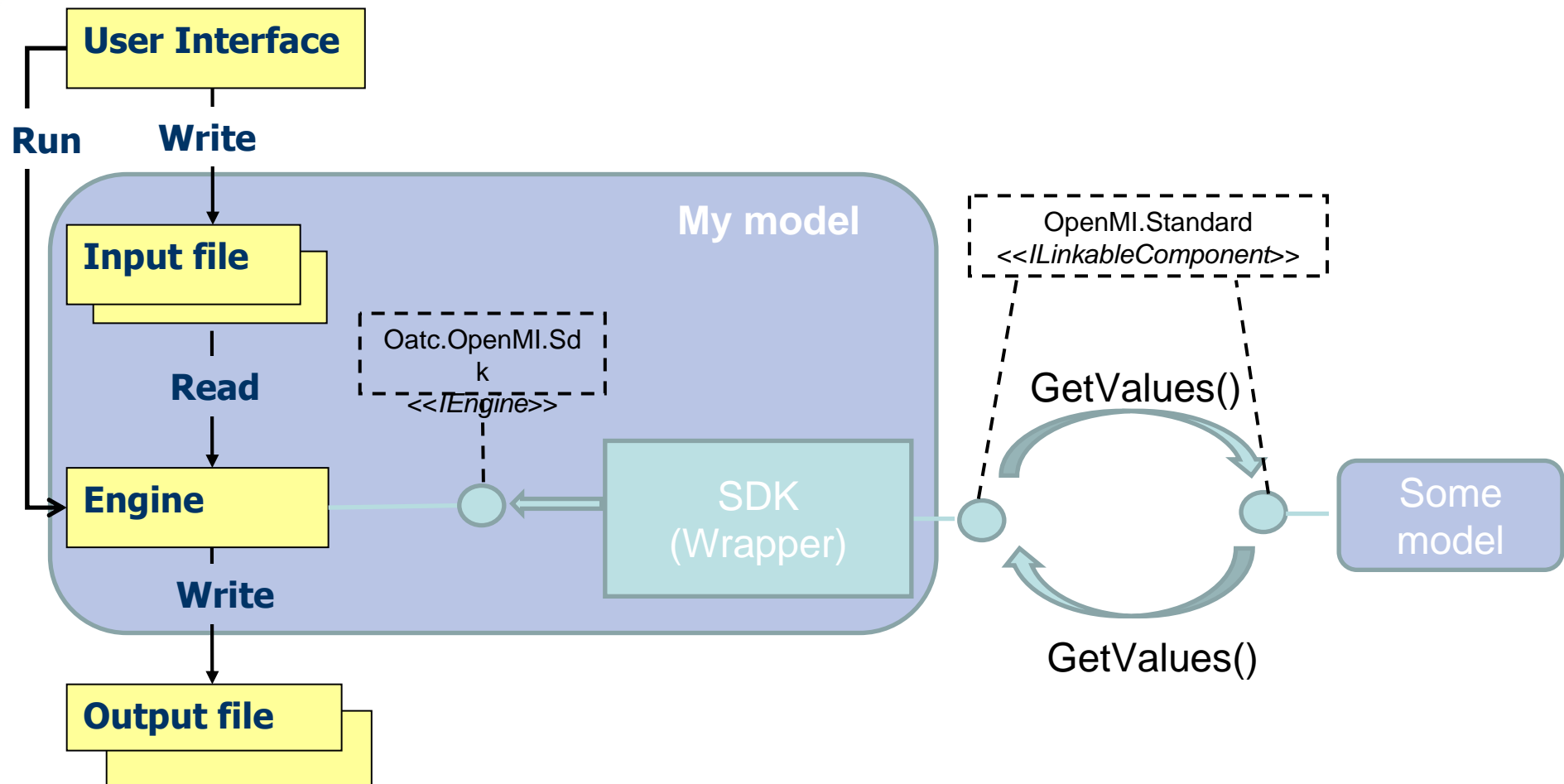
<<IEngine>>

SDK (Wrapper)

GetValues()

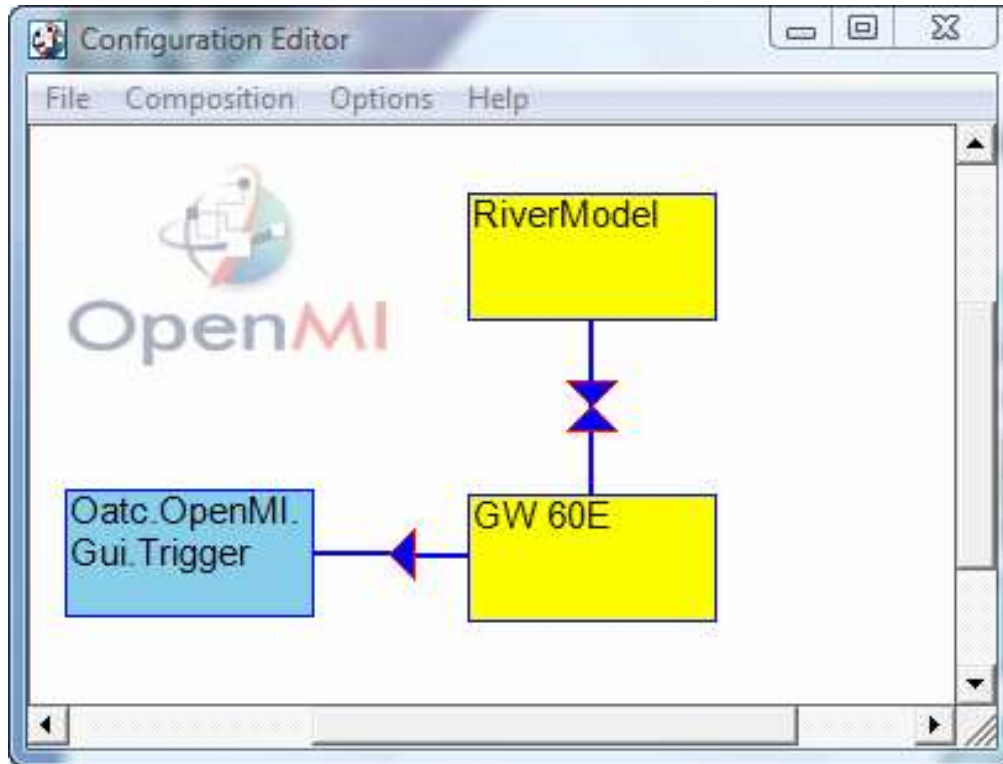
GetValues()

Some model



1. Make sure that your engine is a Dll not a Exe
2. Separate initialization, Perform time step, and finalization
3. Implement IEngine
4. Attach the SDK wrapper
5. Done
6. See also the simple river example

Remarks about the configuration editor

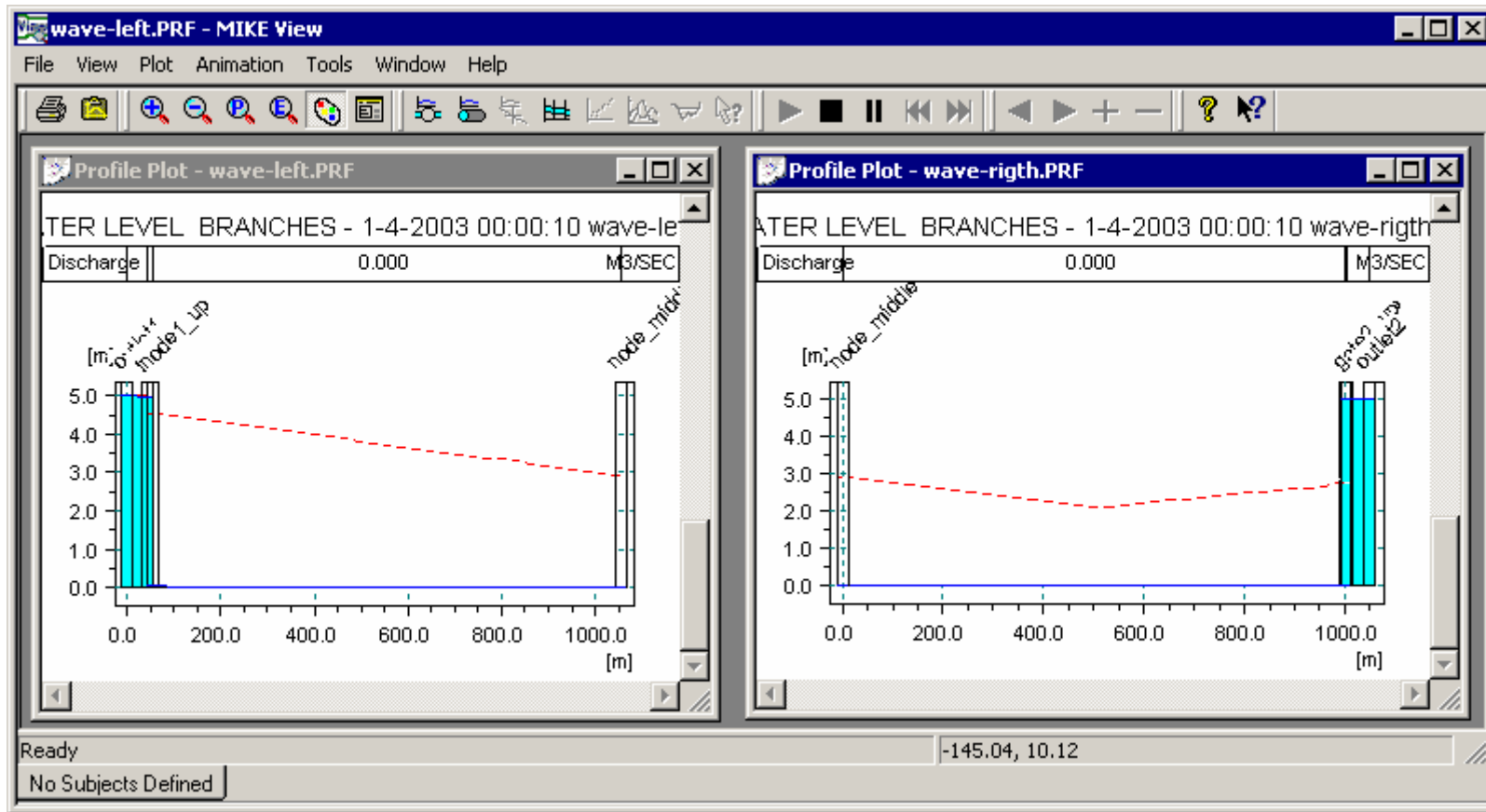


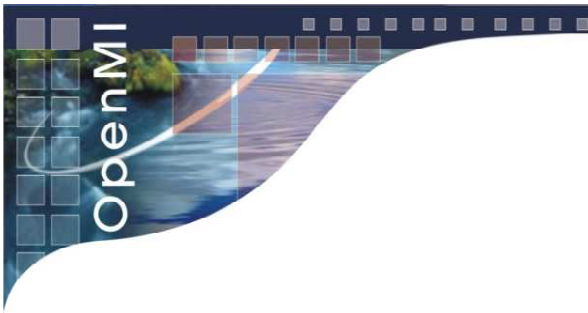
- Enables user to create linked systems and run these
- Is using the OpenMI standard – not part of the standard
- We encourage other people to create more sophisticated GUI's

Final remarks

- The SDK makes model migration easier
- The SDK is aimed at time stepping numerical models
- The SDK is not mandatory to use (only the standard matters)
- SDK and the standard together provides stability and flexibility

Mouse to Mouse





Thank you for you attention

