

Distributed computing

OpenMI components (LinkableComponents) within an OpenMI configuration will always run on the same machine in a single tread. However, if you construct your OpenMI component, using the wrapper pattern (see below), only the wrapper needs to run in the same tread as the other OpenMI components in the OpenMI configuration. The calculation engine, or whatever data source that works in the background can run on a remote machine.

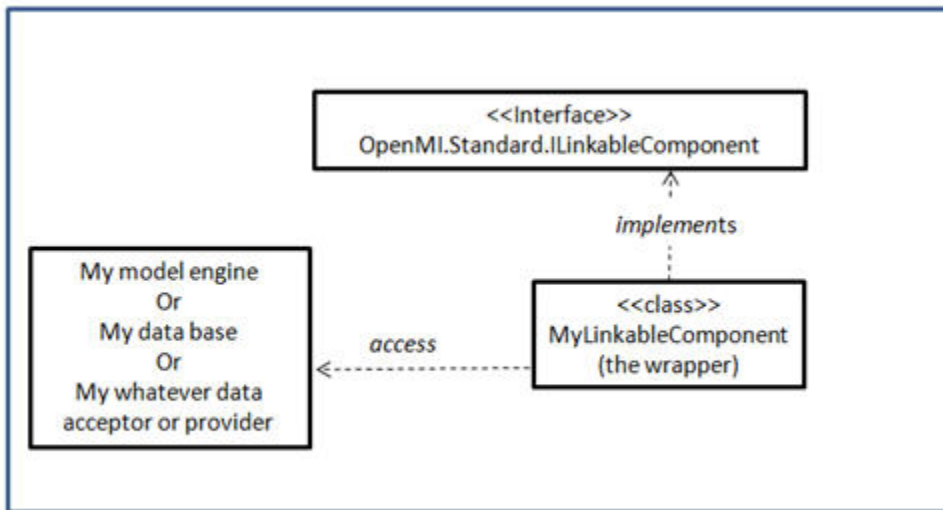


Figure 1: OpenMI wrapper pattern

When you use the OpenMI wrapper pattern you will create a new class which implements the ILinkableComponent interface. This component will access your original component in order get and set data. The trick is that all OpenMI specific things can be located in the wrapper, whereas the original component does not depend on anything related to OpenMI. Moreover, the communication between the wrapper and the original component happens in the private area (nobody's business how this takes place). This means that the original component can be located on a remote server (windows, Linux, or whatever).

On figure 2 below you can see an example of a small configuration with two components and a trigger. The configuration runs on the local machine, and the MyLinkableComponent object is communicating with a component on a remote server.

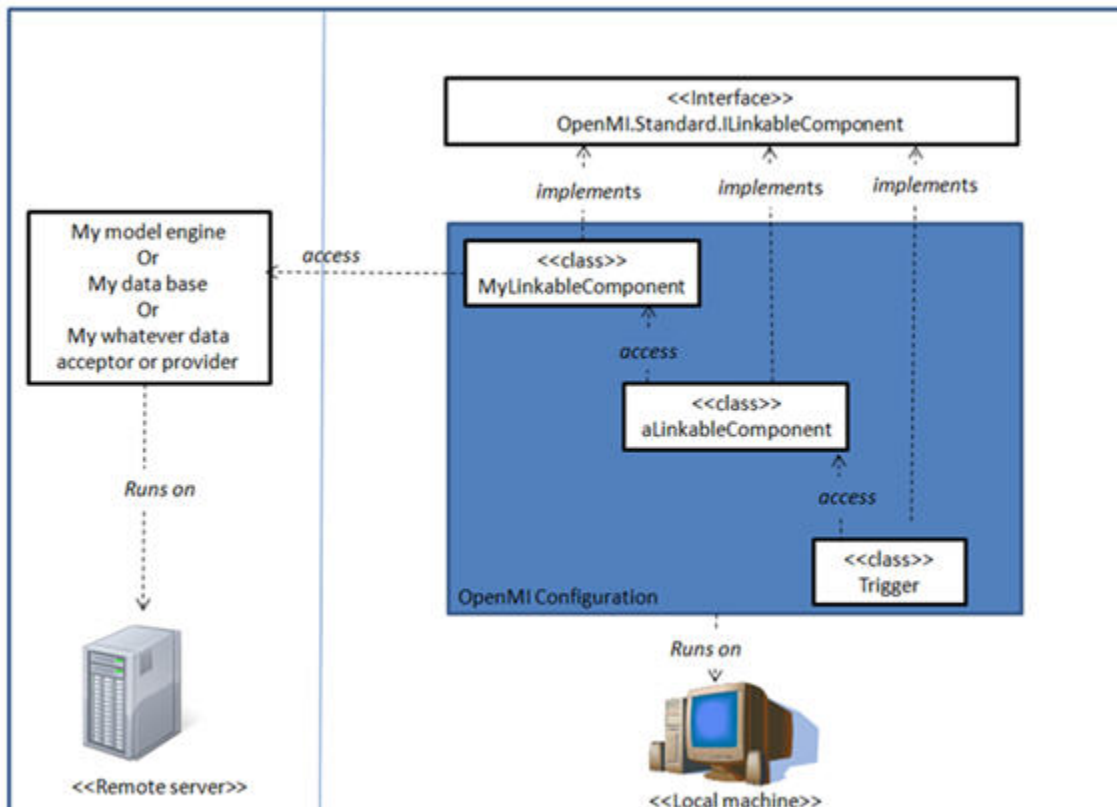


Figure 2: Distributed OpenMI

Say you have a database with time series for precipitation you want to make available through OpenMI. You want to keep your database on your server, but want to enable OpenMI components to access the data from other computers. As explained above, you need to implement a relatively small component/class that implements the `ILinkableComponent` interface. This component will internally have all information and functionality needed in order to retrieve data from your time series database on your server. So, users simply download your `LinkableComponent`, add it to their OpenMI configuration, add links between components in the configuration, and run the configuration, without having to worry about how the data is transferred from the remote precipitation database to the local machine.

One typical question at this point is: "If I run one model engine on a remote machine, will the configuration then run faster?". If you are not doing anything specifically to enable your component to run in parallel, the answer is no. Typically, when `LinkableComponents` returns after `GetValues` calls they will stay idle until they are asked for value again. However, this will depend on how you implement your component. You can allow your remote component to keep running after values have been returned and until data is needed from other `LinkableComponents`. In this case the composition can potentially run faster.