ModelMessageInterface

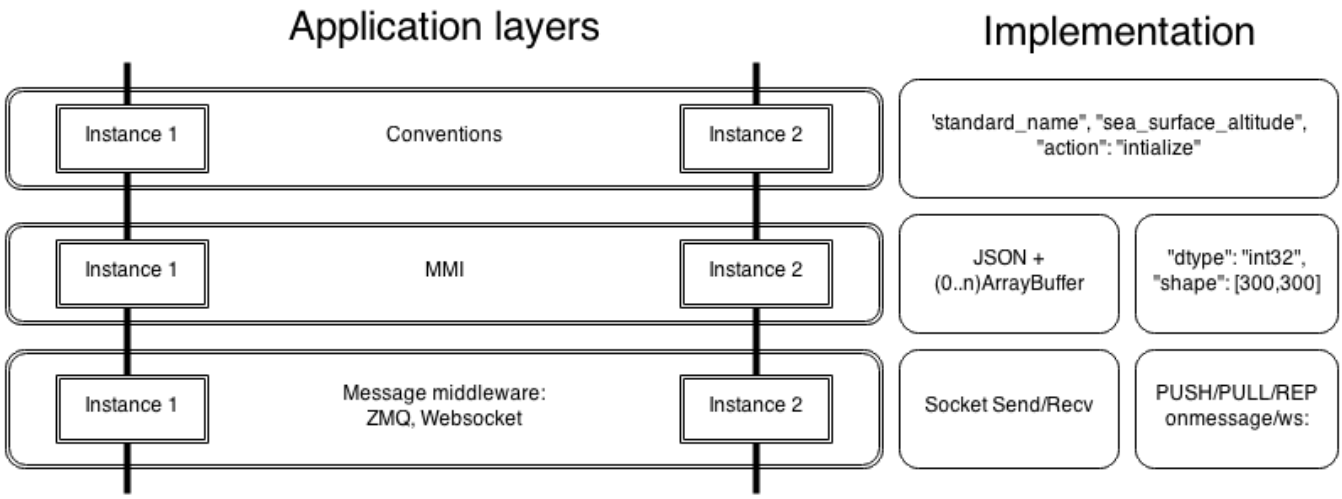
This document describes the Model Message Interface (MMI): a draft protocol for serializing messages between numerical models and between numerical modesl and other programs.

There is already a common protocol in use for model messages, MPI. This works great for communication within a model. Since MPI2 it can also be used to set up ad hoc communication between models.

There are scenarios where communication through MPI is not the most appropriate approach:

- Inter-language communication (e.g. JavaScript and C# support for MPI is unavailable or lagging)
- Flexible process structures (e.g., dynamic populations of distributed programs)
- Communcating through the web (e.g., through firewalls).

Here we describe a serialization protocol that can be used as a layer on top of alternative messaging protocols such as ØMQ and WebSockets.



Our main focus is sending and receiving n-dimensional arrays of simple fixed-length types such as integers and floating-point values, along with metadata and additional attributes. We base our data model on the Variables and Attributes from the Common Data Model [*ref*].

A message contains a block of metadata followed by the data raw, binary format.

Metadata is in JSON format and UTF8 encoded. It contains at least the following three attributes:

```
{
  name: "variable",
  shape: [3,3],
  dtype: "float64"
}
```

With CF extension:

An extended example:

```
{
  name: "variable",
  shape: [3,3],
  dtype: "float64",
  attributes: {
       standard_name: "sea_surface_altiude",
       units: "m"
  },
}
```

With numpy slicing convention:

```
{
  name: "variable",
  shape: [3,3],
  dtype: "float64",
  continuguous: "C",
  strides: [[0,1],[0,2]]
}
```

An intersection of the Python buffer protocol and the JavaScript ArrayBuffer protocol is forseen for the bulk (binary) data transmission.

Implementations

- https://pypi.python.org/pypi/mmi
- https://github.com/openearth/mmi-python
- https://github.com/openearth/mmi-csharp