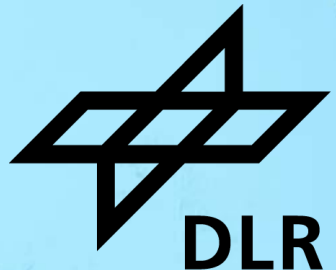


CONNECTING CODES TO ICON VIA THE COMMUNITY INTERFACE (COMIN)

The Modular Earth Submodel System as a first complex ComIn plugin

K. Hartung⁽¹⁾, P. Jöckel⁽¹⁾, A. Kerkweg⁽²⁾, B. Kern⁽¹⁾, and W. Loch⁽³⁾

⁽¹⁾ DLR Institut für Physik der Atmosphäre, ⁽²⁾ FZ Jülich, ⁽³⁾ Deutsches Klimarechenzentrum



ICON Community Interface ComIn



- Standardized **public interface** for third party codes (**'plugins'**) coupled to ICON
- Significantly **reduced maintenance** for ICON as well as for third party code developers
- Plugins **easier to migrate** to new ICON releases
- Establishing ICON as the core model for applications ranging from **NWP** to **ESM**
- Enables **multi-language support** (Fortran, C/C++, Python)



- Organizes data exchange and simulation events between ICON and plugins
- ComIn Callback Register and ComIn Library Adapter
- https://www.icon-model.org/icon_model/community-interface

Modular Earth Submodel System (MESSy)

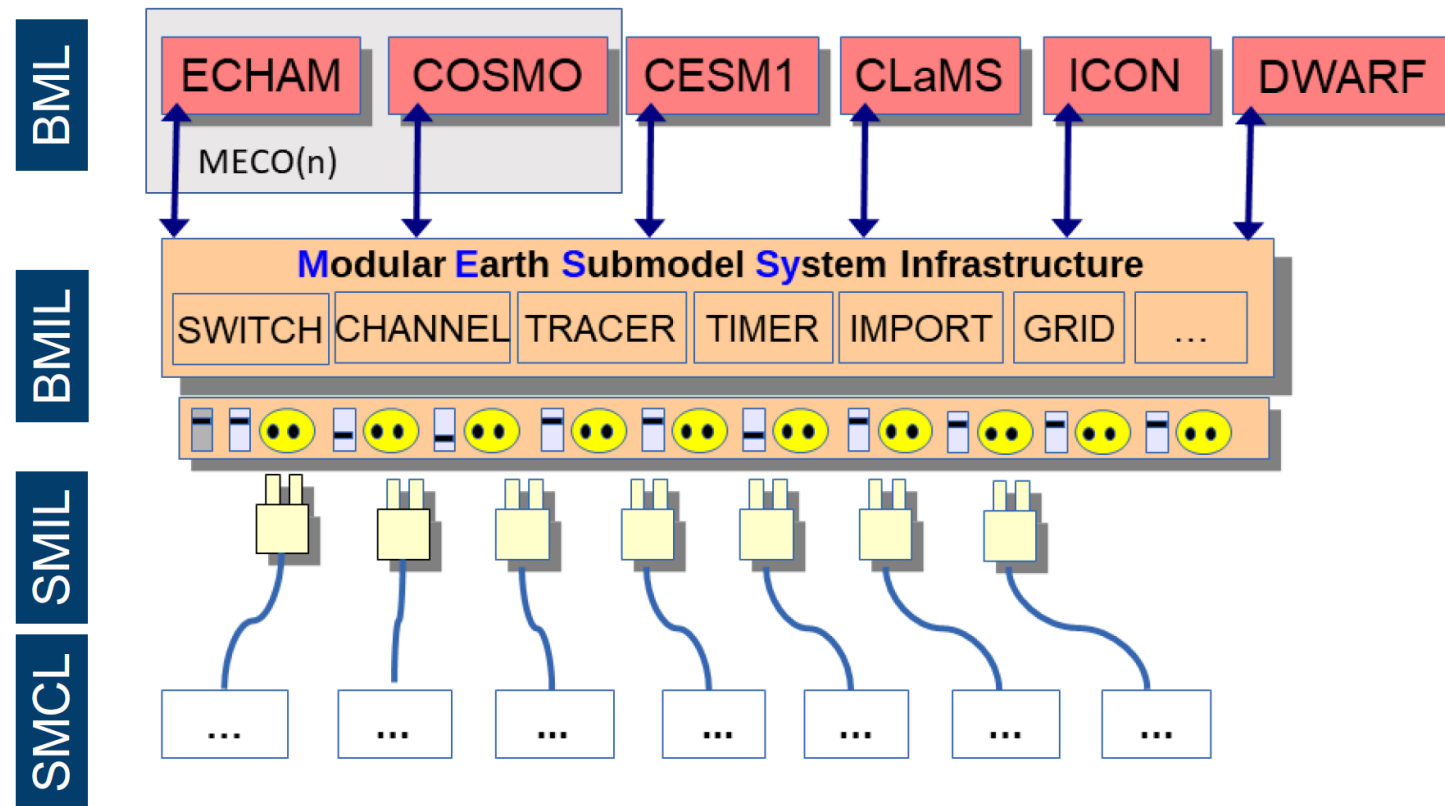


- Framework to couple scientific codes to numerical weather prediction and climate models
 - > 20 partner institutes
- <https://www.messy-interface.org/>

Modular Earth Submodel System (MESSy)



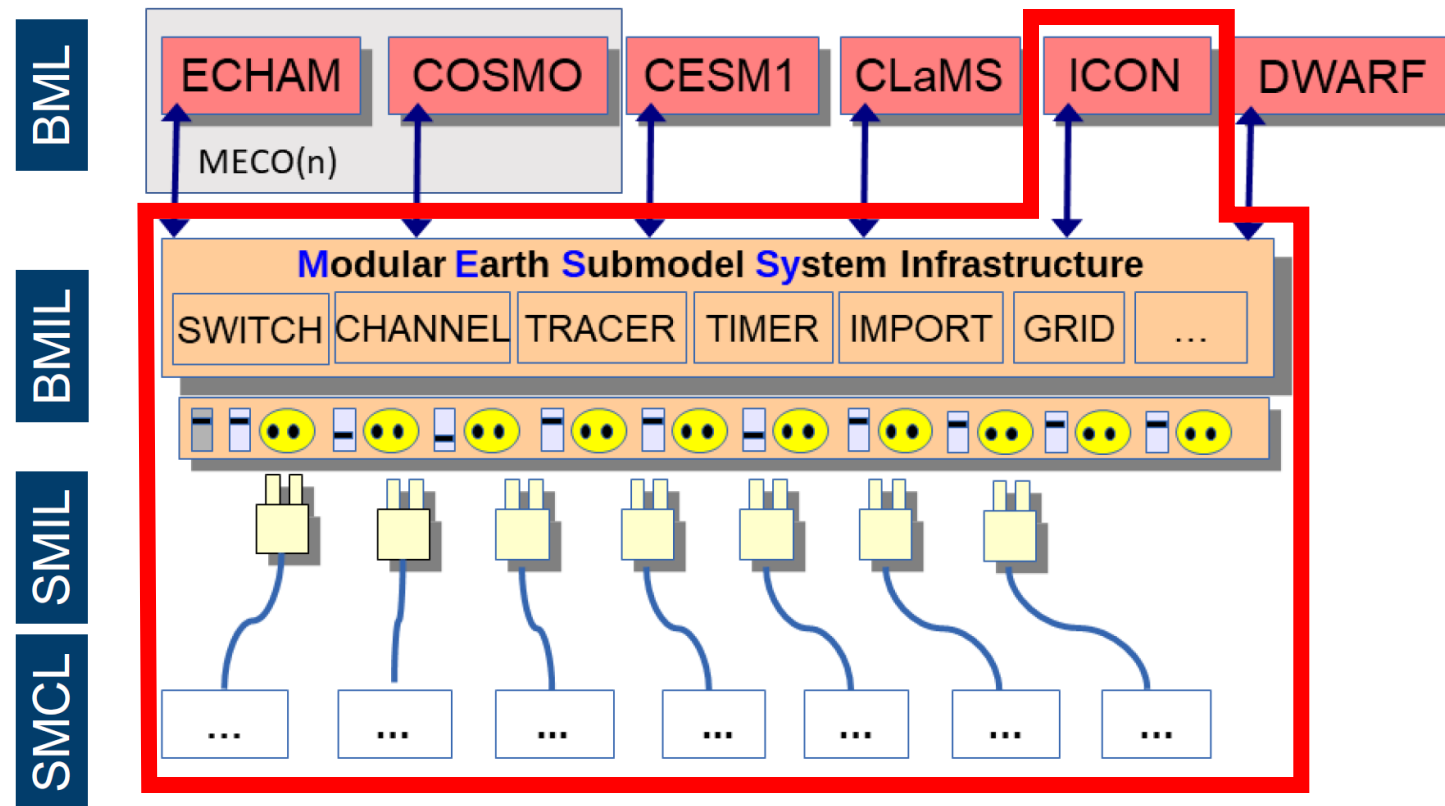
MESSy code structure



- Framework to couple scientific codes to numerical weather prediction and climate models
 - > 20 partner institutes
- <https://www.messy-interface.org/>

Modular Earth Submodel System (MESSy)

MESSy code structure

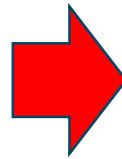
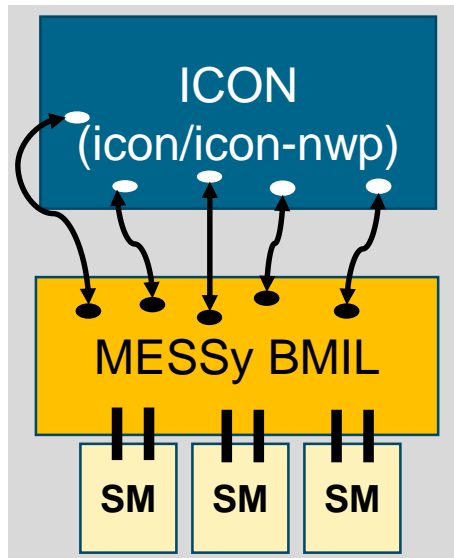


- Framework to couple scientific codes to numerical weather prediction and climate models
 - > 20 partner institutes
- <https://www.messy-interface.org/>

Motivation

Modular Earth Submodel System Current status

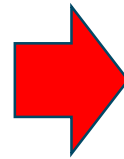
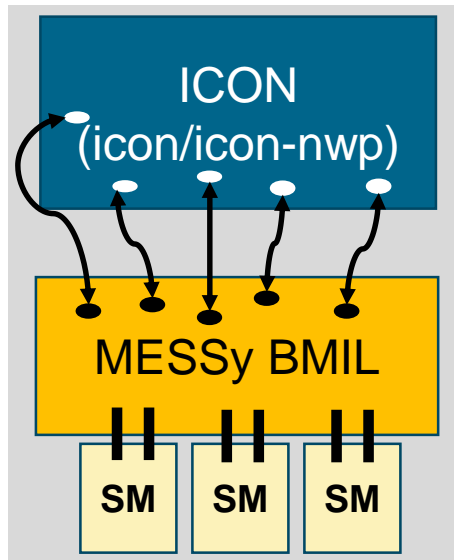
Goal: reduced effort to keep up-to-date with ICON developments



Motivation

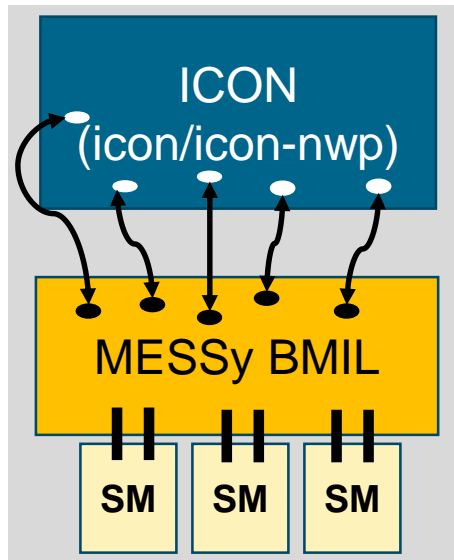
Modular Earth Submodel System Current status

Goal: reduced effort to keep up-to-date with ICON developments

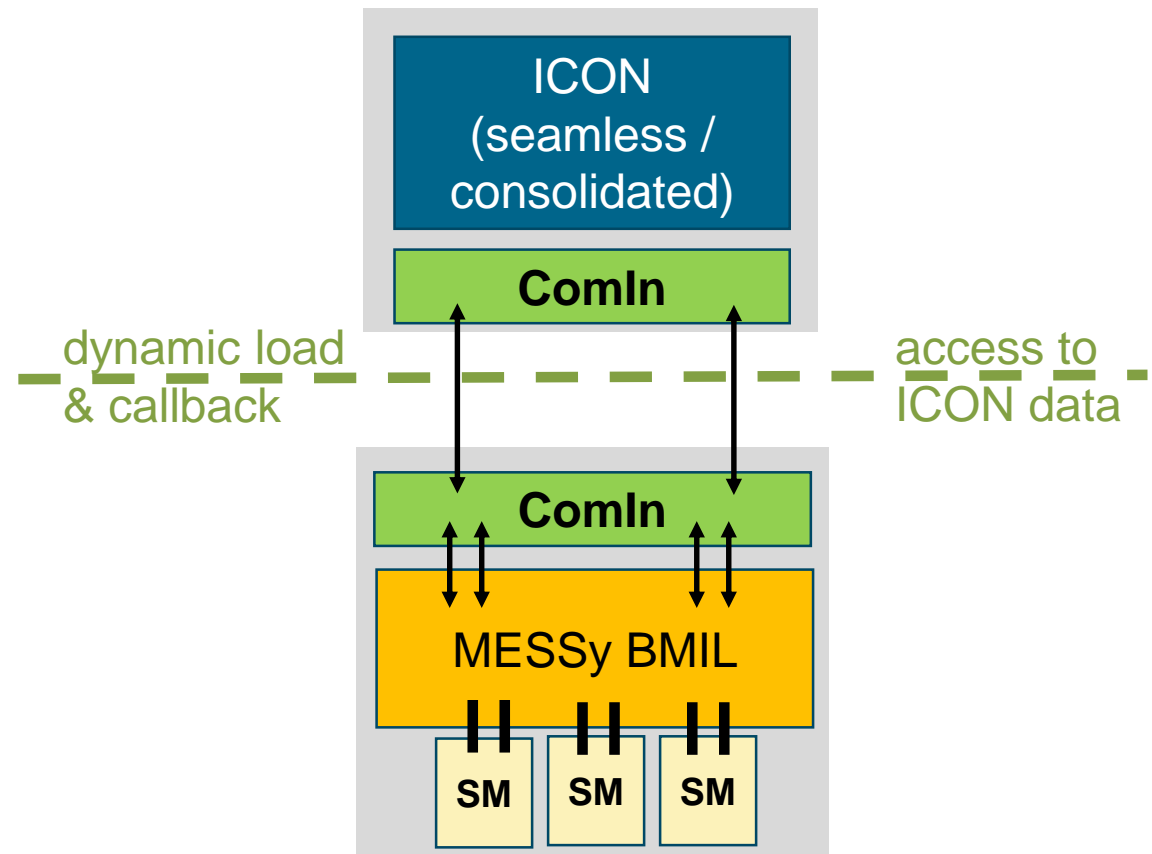


Motivation

Modular Earth Submodel System Current status



Goal: reduced effort to keep up-to-date with ICON developments



natESM sprint: Couple MESSy to ICON via the ICON Community Interface (ComIn)



Bundesministerium
für Bildung
und Forschung

- <https://www.nat-esm.de/>
- Open call for proposals from model-development groups across Germany
- Sprints are focused on technical objectives, flexible, tailored to research goals and timelines
- Up to 6 months, in-depth partnership between applicant and Research Software Engineers (RSE)

natESM sprint: Couple MESSy to ICON via the ICON Community Interface (ComIn)



- 07/2023 – 01/2024
- Implement ComIn in MESSy, advance towards a working setup
- Co-existence of original and new implementation during development
- Document steps and challenges for other plugins



Bundesministerium
für Bildung
und Forschung

- <https://www.nat-esm.de/>
- Open call for proposals from model-development groups across Germany
- Sprints are focused on technical objectives, flexible, tailored to research goals and timelines
- Up to 6 months, in-depth partnership between applicant and Research Software Engineers (RSE)

Implementation approach



- Iterative implementation updates
 - Testing functionality of intermediate steps
 - Workarounds/updates if either MESSy or ComIn do not support functionality at the moment
 - Direct feedback to ComIn development
 - Shaping future ComIn version: short term and long term development goals
-
- ComIn is a lightweight interface
 - ComIn provides grid and decomposition information, variable meta-data, ...
 - ComIn provides a MPI communicator but no communication patterns

Results from the implementation

1. Getting started and first tests

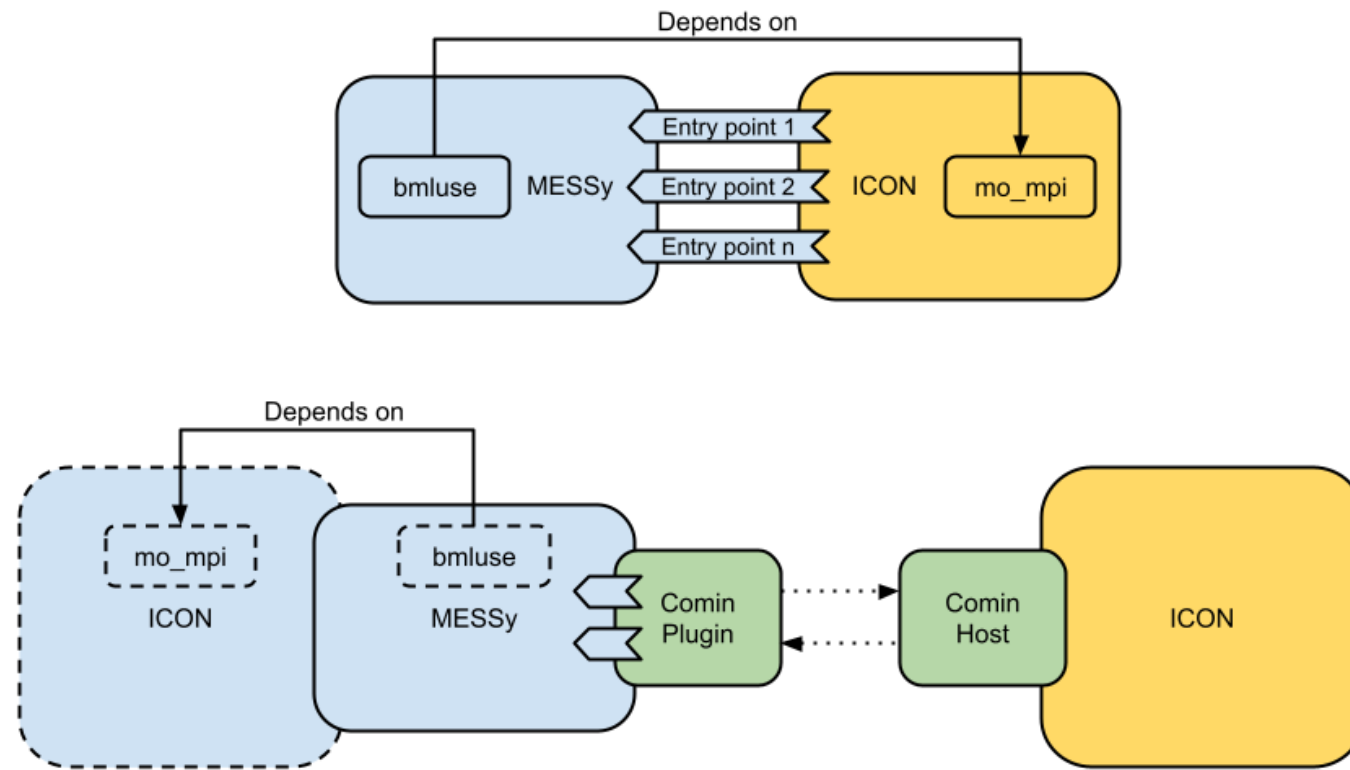


Diagram by Wilton Loch

Results from the implementation



2. Porting model initialization to ComIn

```
SUBROUTINE messy_comin_setup()
```

```
CALL init_icon_get_mpi()
```

some workarounds

```
CALL comin_callback_register(EP_SECONDARY_CONSTRUCTOR,  
    messy_comin_constructor, ierr)
```

register callbacks

```
CALL comin_callback_register(EP_ATM_INIT_FINALIZE,  
    messy_comin_atm_finalization, ierr)
```

```
CALL messy_setup()
```

initialize MESSy and MESSy submodels

```
CALL messy_initialize
```

```
CALL messy_new_tracer
```

gather info on new tracers, receive descr. data

```
CALL messy_request_tracers
```

prepare MESSy tracer metadata and
call comin_var_request_add

```
END SUBROUTINE messy_comin_setup
```

Results from the implementation



2. Porting model initialization to Comln

```
SUBROUTINE messy_comin_constructor()
```

```
CALL messy_get_tracer_metadata_comin
```

receive tracer metadata

```
CALL messy_init_memory
```

receive pointers to ICON variables and tracers, init SM memory

```
END SUBROUTINE messy_comin_constructor
```

```
SUBROUTINE messy_comin_atm_finalization()
```

```
CALL messy_init_coupling
```

incomplete because MPI not fully set up

```
CALL messy_read_restart
```

```
CALL messy_init_tracer
```

read restart and initialize tracer fields

```
END SUBROUTINE messy_comin_atm_finalization
```

Results from the implementation



3. Feedback from first complex plugin

Valuable feedback from first complex plugin for (early stage) ComIn development, e.g.:

- metadata expanded and access/set routines added
- all fields at cell centres are shared via ComIn, not just 3D and surface fields
- convenience function to retrieve time step of each domain
- some additional descriptive data

4. Additional updates within MESSy

- enable building MESSy as shared library (partly done before sprint)
- workaround in MESSy to not require hybrid vertical coordinate from ICON
- start preparation of YAXT backend in MESSy to provide MPI functionality
- update MESSy run-script to work with ComIn

Outlook and open questions (from the MESSy and natESM sprint perspective)

final steps of initialization

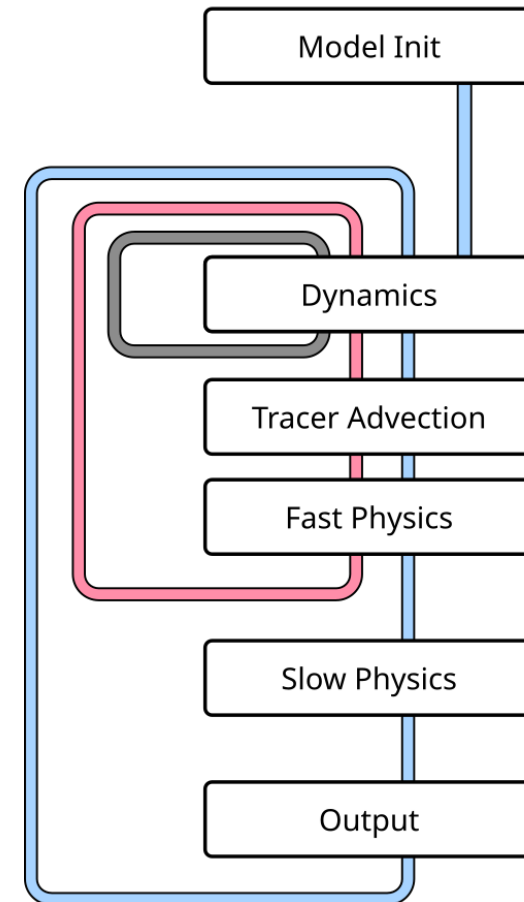
- set up MPI with YAXT

using ComIn during time loop

- add calls to current entry points in MESSy
- some workarounds, e.g. for access to currently (sub)routines-local fields, masking regions for parameterizations

recommendations for community

evaluation of implementation



Outlook (ICON Community Interface, ComIn)



- ComIn has extensive documentation, examples and tests
 - Documentation of constraints and workarounds
 - Recommendations in form of sprint documentation, collecting best practices, expertise on ComIn of natESM RSEs
 - Community feedback on the ComIn development
 - Identify “shortcomings”, suggest and support future extensions
 - ... ComIn v0.2.0 ...
-
- ComIn is well-documented, flexible, lightweight, ready to use
 - Some knowledge of ICON internals are needed
 - Fit to purpose, with some constraints...

Topic: **Connecting codes to ICON via the Community Interface (ComIn)**
The Modular Earth Submodel System as a first complex ComIn plugin

Date: 2023-03-05

Author: Bastian Kern

Institute: Institut für Physik der Atmosphäre, DLR e.V.

Image credits: All images “DLR (CC BY-NC-ND 3.0)” unless otherwise stated