Sprint 15
MESSy-ComIn2
ComIn Integration Time Loop

Bastian Kern, Kerstin Hartung, Patrick Jöckel, Astrid Kerkweg (DLR-PA, FZJ IEK-8)

Lakshmi Aparna Devulapalli, Aleksandar Mitic (DKRZ)

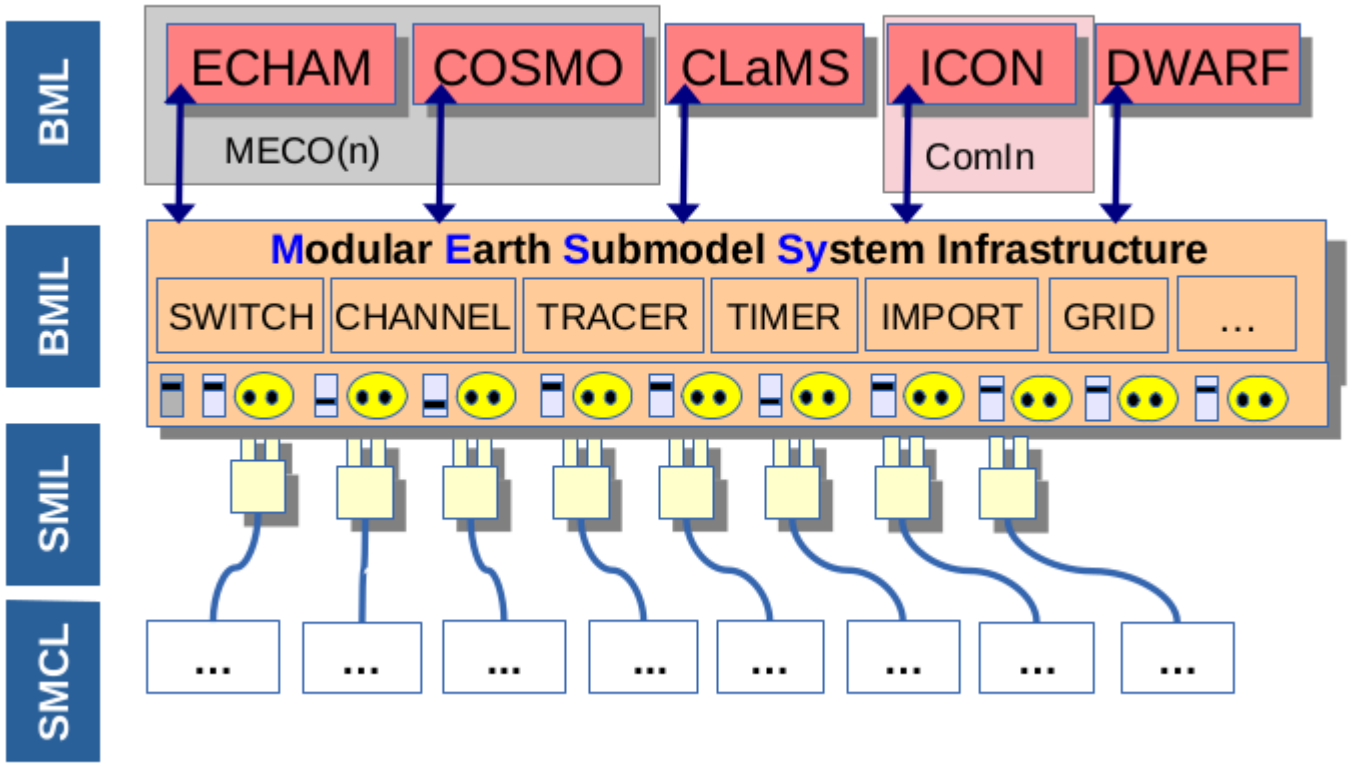
natESM

DLR

JÜLICH Forschungszentrum

DKRZ DEUTSCHES KLIMARECHENZENTRUM

Modular Earth Submodel System (MESSy)



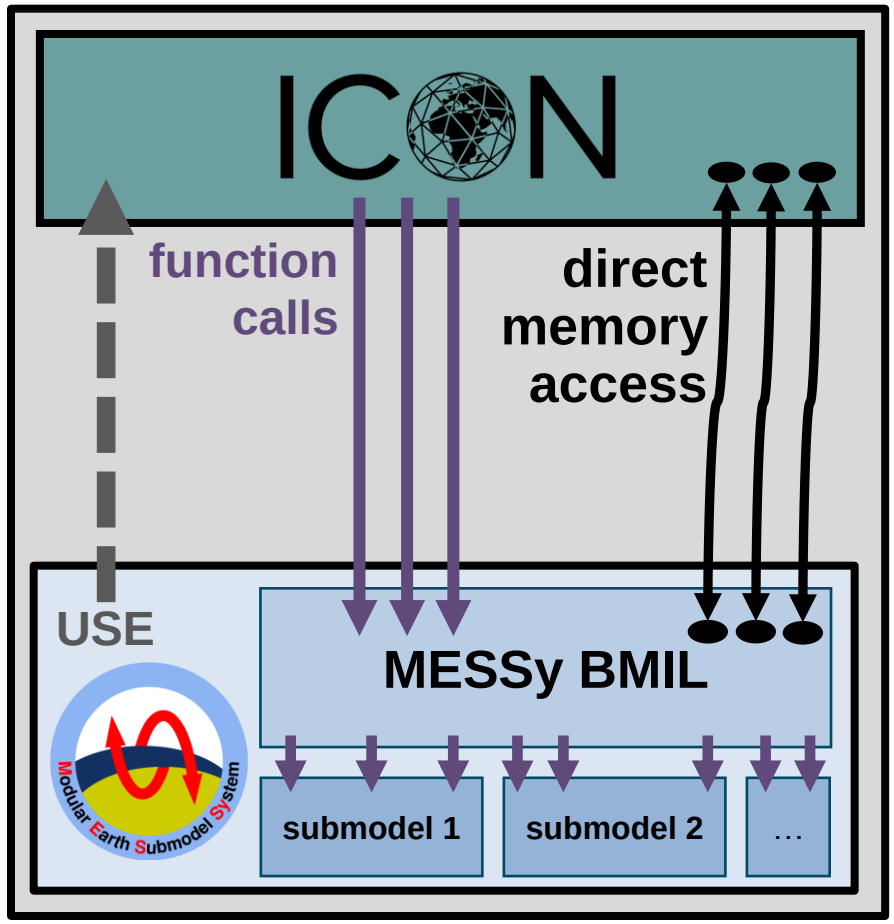
- Software framework for coupling of scientific codes to numerical weather forecast and climate models
  - > 20 institutions
- https://www.messy-interface.org/

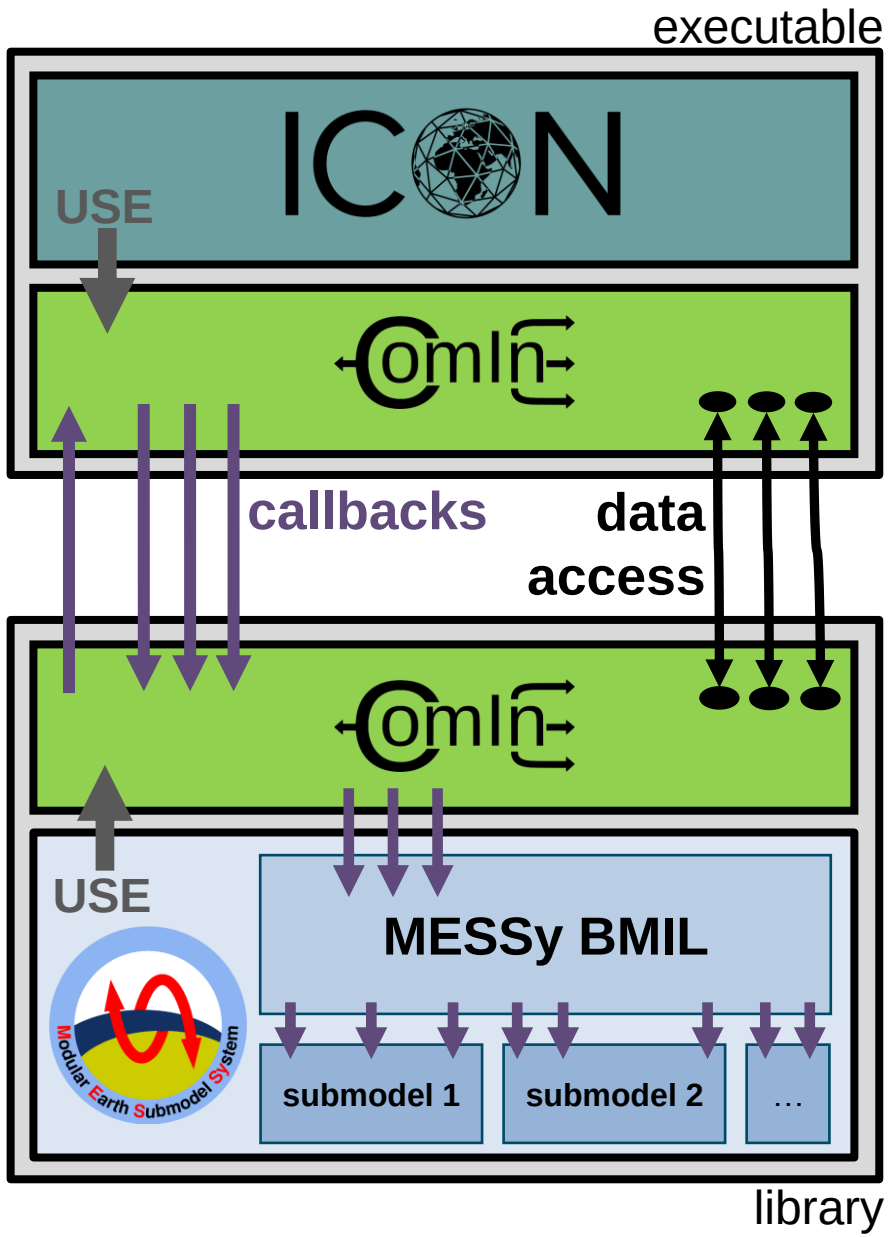
ComIn – ICON Community Interface
Plugin Interface in ICON

- Software library for the coupling of plugins to ICON
  - Pre-defined entry points and registry for callback functions
  - Access to model variables and metadata
  - Creation of new variables
- No code dependencies between ICON and plugins
- Open source release, distribution with ICON model code, extensive documentation
- Implementation of MESSy as ComIn plugin

https://docs.icon-model.org/tools/comin/comin.html

ComIn: Community Interface – Plugin Interface in ICON





Objectives of the natESM sprint (I)

September 2024 - March 2025 (6 months)
Responsible RSE: Lakshmi Aparna Devulapalli, DKRZ
Responsible scientist: Bastian Kern, DLR-PA

- Automatic channel-object creation
  - Access to ICON variables and metadata via ComIn
  - Connect MESSy metadata and memory in an automated way
  - Creation of dimensions, representations and channel objects for all exposed variables
- Dimensional Semantics (added during the sprint)
  - Improve metadata of ICON
  - Provide information describing dimensions of variable fields
- YAXT communication patterns - Gather, scatter, boundary exchange
  - Remove dependencies of MESSy on parallel communication methods of ICON
- MPI communication – one-to-one and all-to-all communication
  - Remove dependencies of MESSy on MPI communication routines of ICON

Objectives of the natESM sprint (II)

September 2024 - March 2025 (6 months)
Responsible RSE: Lakshmi Aparna Devulapalli, DKRZ
Responsible scientist: Bastian Kern, DLR-PA

- ComIn callback calls for entry points in time loop
  - Extend ComIn callback calls to cover MESSy entry points in the ICON time loop
- Resolving further code dependencies
- Providing access to “local fields”
- Evaluation
- Documentation

Automatic creation of MESSy Channel Objects

- Automatic creation in a generic way
  - Replacing the existing error-prone algorithm (“if statements”)
- Generalized naming nomenclature for dimensions and representation
  - Dimensions and representations: geometric structure of the data

→ Loop over all the fields being exposed by ICON/ComIn

  - Create generic names for dimension and representation based on the geometric information provided by ICON/ComIn
  - Check if the dimension has to be created
  - Check if the corresponding representation has to be created
  - Create a new channel object (use dimension id and representation id)

→ This depends on accurate geometric information received from ICON/ComIn! → dimensional semantics

Dimensional semantics (*)
Basics

- ICON/ComIn interface provides a pointer to 5D-memory
- Not all variable fields have a “standard” ordering of dimensions
  - Which dimension represent the horizontal, vertical, or other extend?
  - Heuristic determination usually breaks for ICON fields, which do not follow “standard” dimensional ordering
- “Dimensional semantics” were proposed as additional metadata for variables
  - Presented during the ICON all hands meeting in October 2024, received positive feedback
  - Implementation of “dimensional semantics” is still in progress

(*) This task was added during the sprint

Dimensional semantics (*) Implementation

→ Interface modification in ComIn to receive dimensional semantics
- Position indices replaced by dimensional semantics; array `dim_semantics`

→ Adapting MESSy code to utilize dimensional semantics provided by ComIn
- Automatic channel object generation algorithm

→ Implementing the changes in ICON (MR in progress)
- `add_var()` subroutine with optional argument `dim_semantics`

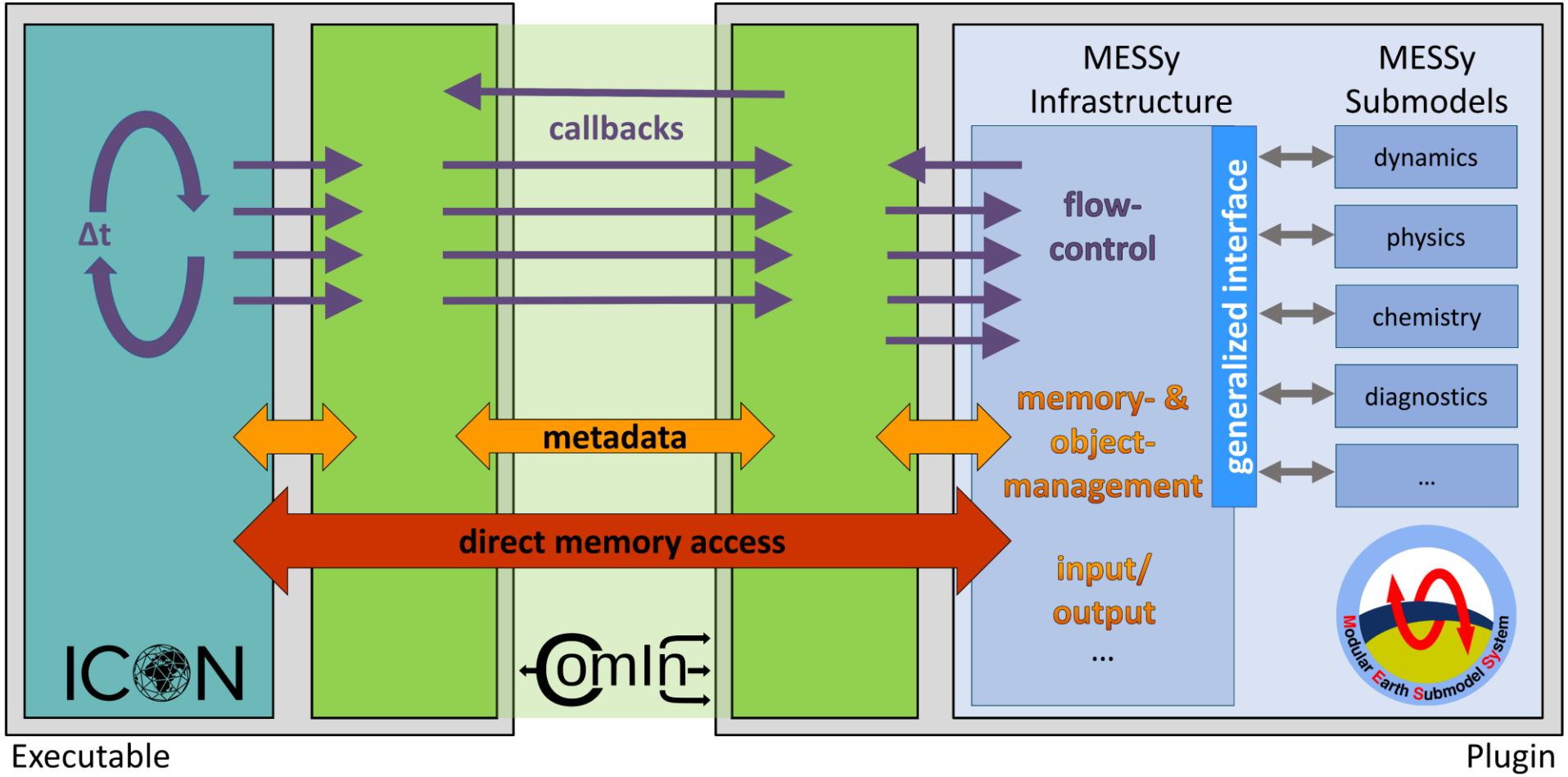
→ Removing the heuristic from ComIn (To be done)

(*) This task was added during the sprint

YAXT, MPI and time loop

- YAXT Communication Patterns
  - One-to-all and all-to-one communication methods in MESSy for data transpositions
  - Remove code dependencies in MESSy
  - Some additional refinement in progress after sprint, exchange with YAXT developers
- MPI – all-to-all communication (support by A. Mitic, ESiWACE)
  - ICON-independent MPI interface for MESSy
  - Point-to-point, collective, asynchronous communication based on workgroup communicators provided by the base models
- Calls inside time loop
  - Additional entry points in ComIn

MESSy as ComIn Plugin



Outlook and open tasks

- YAXT based data transposition
  - Implemented, optimisation required
  - Boundary-exchange work in progress
  - Performance issues with Intel compiler
  - Feedback to / exchange with developers
- Access to „local loop“ variables of ICON
  - Implemented
  - Patching of ICON needed

Conclusion

- Main goals of sprint reached
  - Variable access and channel objects
  - MPI and YAXT communication
  - Remove code dependencies in MESSy
- Organizational
  - Weekly sprint meetings, chat platform
  - gitlab.dkrz.de, issues, CD/CI pipeline for MESSy
  - Several workshops, hackathons
- Support in development, „view from outside“
  - Improvement of own code
  - Feedback & improvements of other codes (ComIn, ICON, YAXT)

Sprint report available:

https://www.nat-esm.de/

Imprint

Topic: **Sprint 15 – MESSy-ComIn2**

Date: 2026-02-24

Author: Bastian Kern

Institute: DLR-PA