7. WMO Data Assimilation Symposium 2017, Florianopolis, Brazil

# Building a Scalable Ensemble Data Assimilation System for Coupled Models

Lars Nerger, Dmitry Sidorenko

Alfred Wegener Institute for Polar and Marine Research
Bremerhaven, Germany

ESM Advanced Earth System Modelling Capacity

ALFRED-WEGENER-INSTITUT
HELMHOLTZ-ZENTRUM FÜR POLAR-
UND MEERESFORSCHUNG

# Motivation

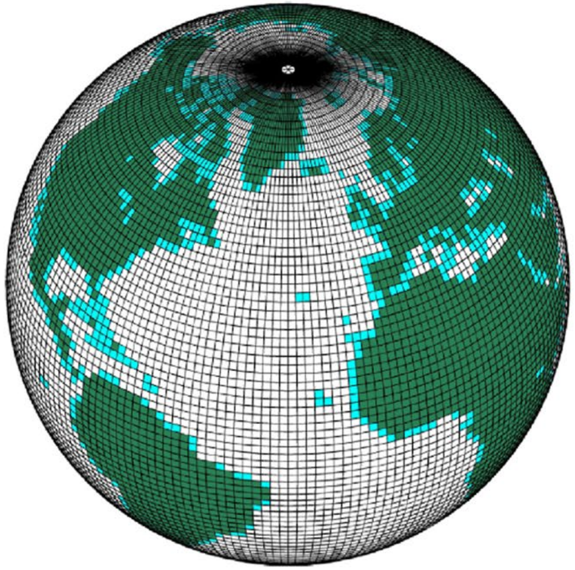
How to build an efficient data assimilation system
– in a simple way?

1. Extend model to integrate an ensemble
    - mainly: adapt parallelization
2. Add analysis step to the model
    - just an update in between time steps

Here discussed for a coupled model

# Example: ECHAM6-FESOM (AWI-CM)

Atmosphere

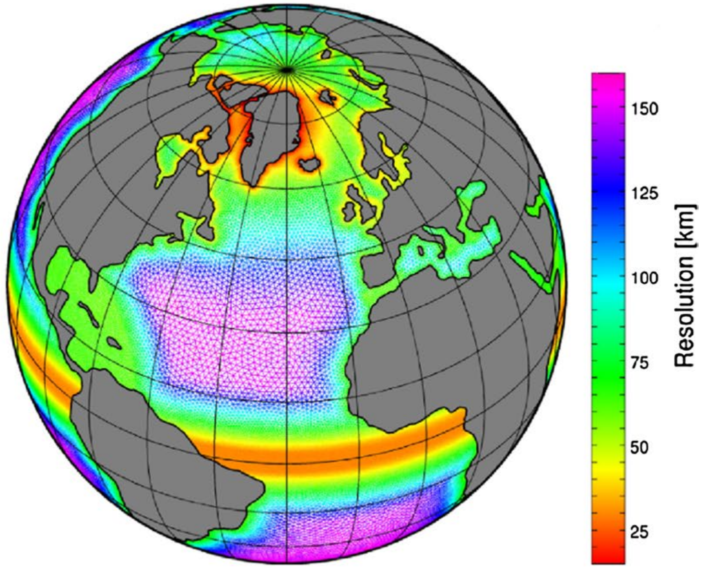


**OASIS3-MCT**

fluxes →

← ocean/ice state

Ocean




**Atmosphere**
- ECHAM6
- JSBACH land

**Ocean**
- FESOM
- includes sea ice

**Coupler library**
- OASIS3-MCT

Two separate executables for atmosphere and ocean

D. Sidorenko et al., Clim. Dyn. 44 (2015) 757

# PDAF: A tool for data assimilation

PDAF Parallel Data Assimilation Framework

## PDAF - Parallel Data Assimilation Framework

- a program library for ensemble data assimilation
- provide support for parallel ensemble forecasts
- provide fully-implemented & parallelized filters and smoothers (EnKF, LETKF, NETF, EWPF … easy to add more)
- easily useable with (probably) any numerical model (applied with NEMO, MITgcm, FESOM, HBM, TerrSysMP, …)
- run from laptops to supercomputers (Fortran, MPI & OpenMP)
- first public release in 2004; continued development
- ~250 registered users; community contributions

Open source:
Code, documentation & tutorials at

http://pdaf.awi.de

L. Nerger, W. Hiller, Computers & Geosciences 55 (2013) 110-118

AWI

# PDAF: User-friendliness

PDAF Parallel Data Assimilation Framework

Assumption: Users know their model

➜ let users implement DA system in model context

For users, model is not just a forward operator

➜ let users extend they model for data assimilation

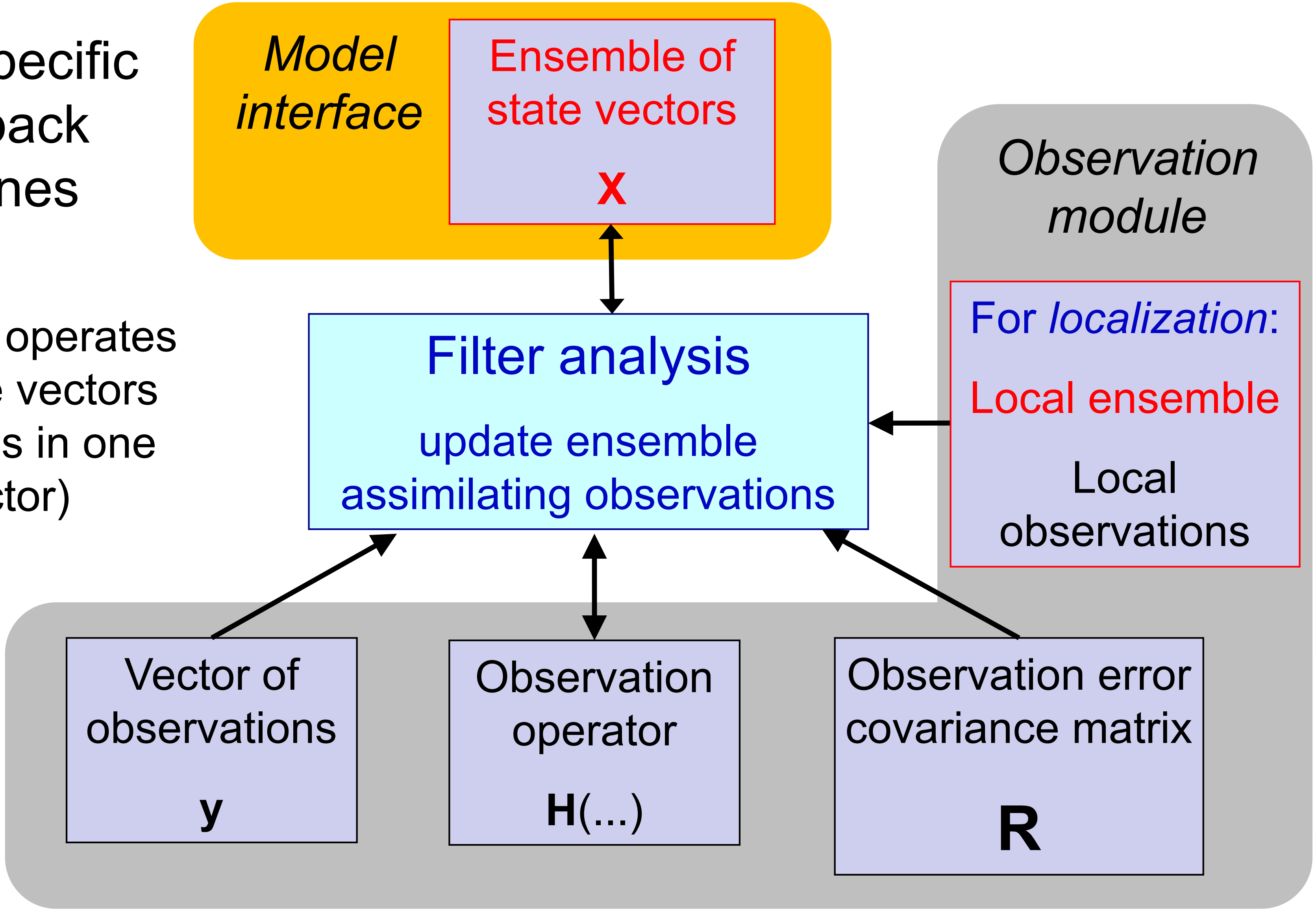
Keep simple things simple:

- Define subroutine interfaces to separate model and assimilation based on arrays
- No object-oriented programming
(most models don't use it; most model developers don't know it; not many objects would be involved)
- Users directly implement observation specific routines
(no indirect description of e.g. observation layout)

# Ensemble Filter Analysis Step

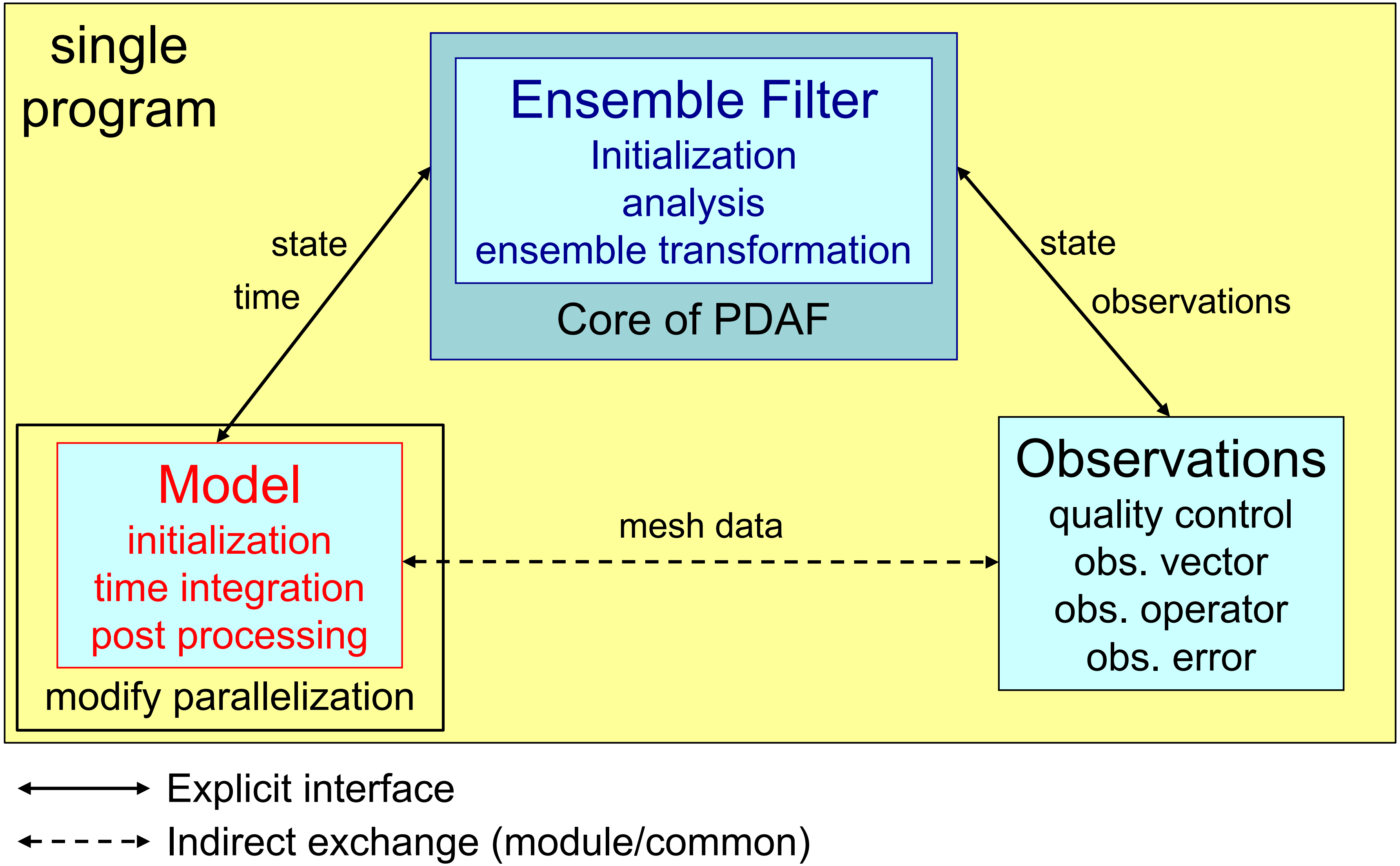
case-specific
call-back
routines

Analysis operates on state vectors (all fields in one vector)



# Logical separation of assimilation system






Nerger, L., Hiller, W. Software for Ensemble-based DA Systems – Implementation and Scalability. Computers and Geosciences 55 (2013) 110-118

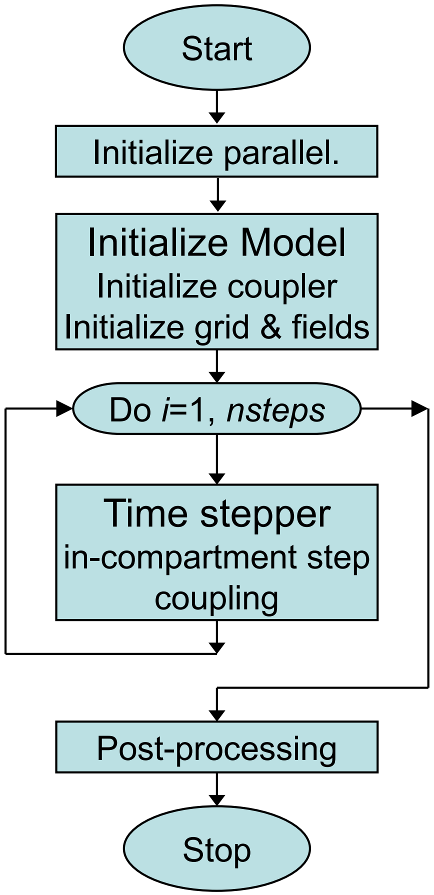
# Extending a Model for Data Assimilation

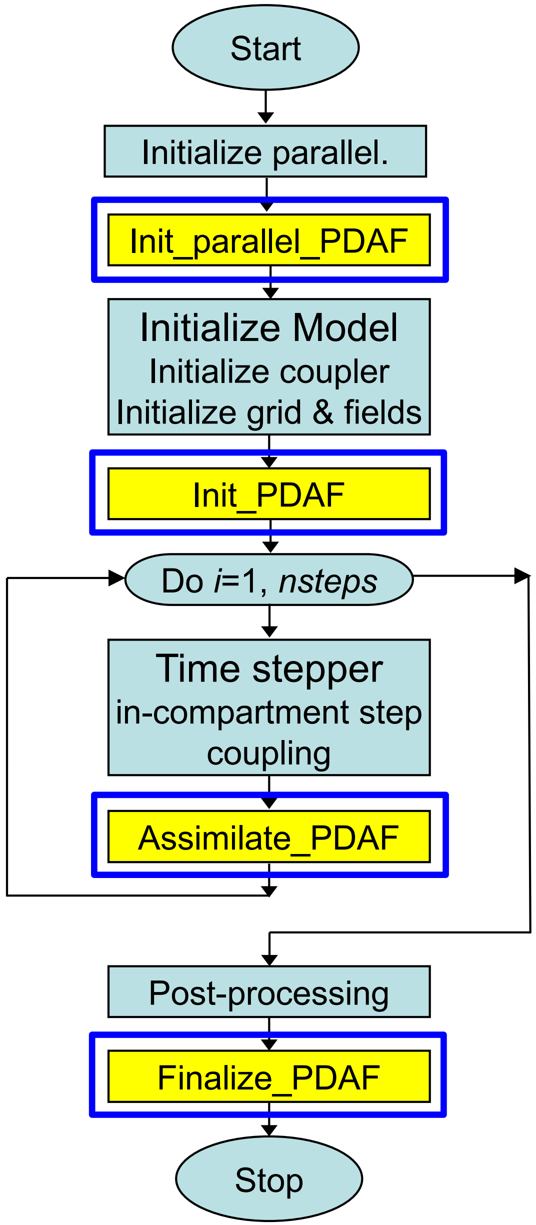


**Model**

*single or multiple executables*

*coupler might be separate program*



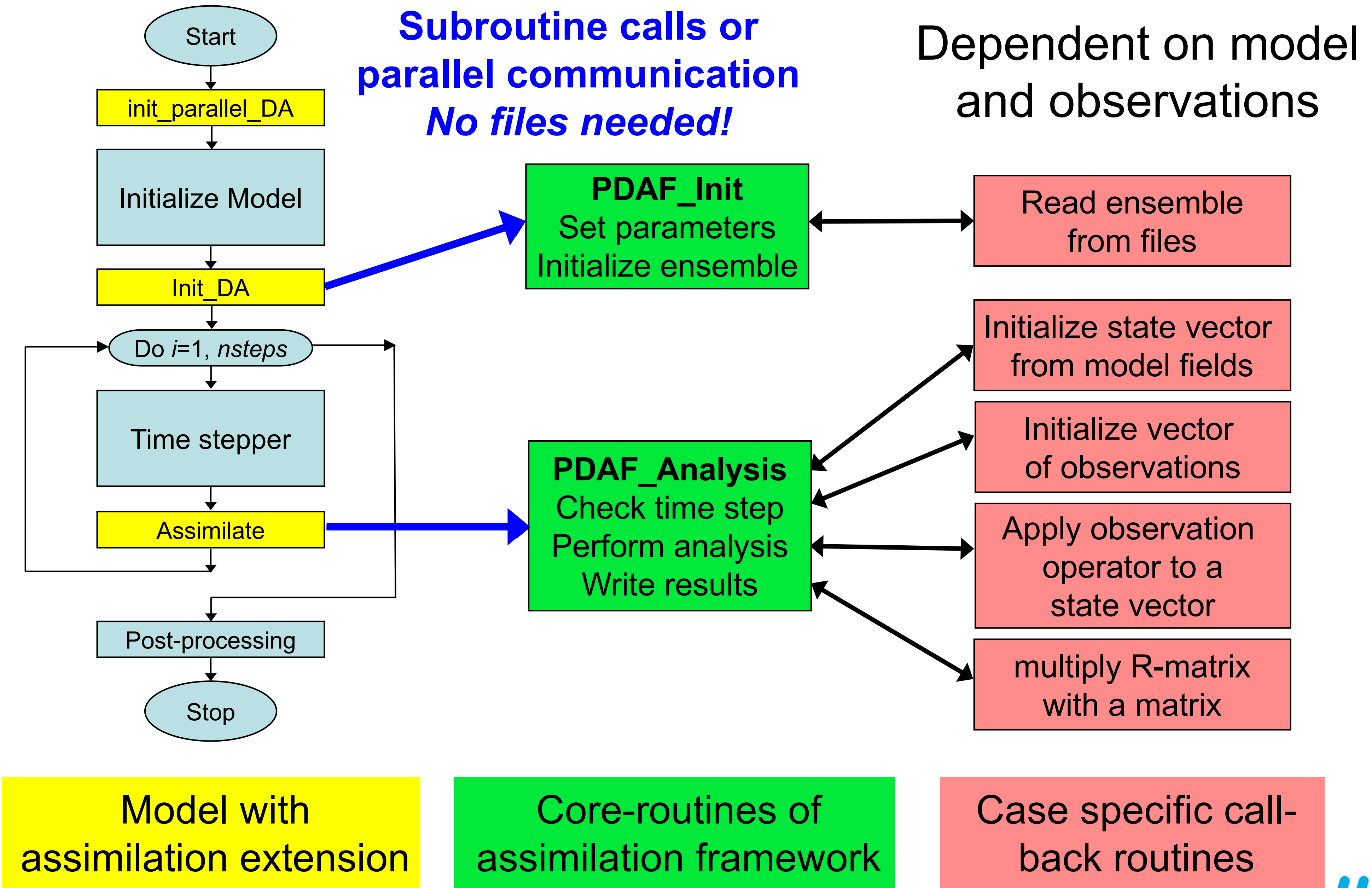

Extension for data assimilation

*plus:*
Possible model-specific adaption

AWI-CM: adaption of coupler (e.g. OASIS3-MCT)

revised parallelization enables ensemble forecast

# Framework solution with generic filter implementation



# Simple Subroutine Interfaces

Example: observation operator

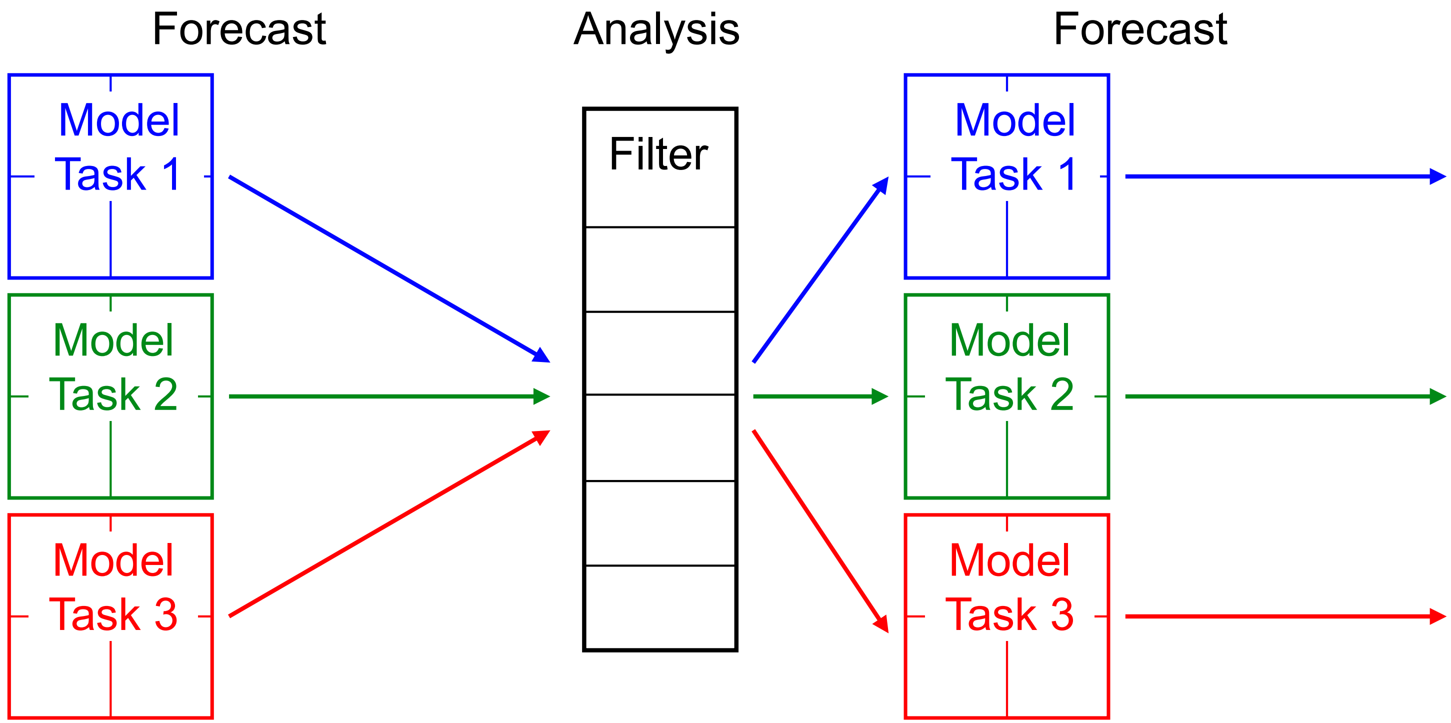
```
SUBROUTINE obs_op(step, dim, dim_obs, state, m_state)

  IMPLICIT NONE

! ARGUMENTS:
  INTEGER, INTENT(in) :: step             ! Current time step
  INTEGER, INTENT(in) :: dim              ! PE-local dimension of state
  INTEGER, INTENT(in) :: dim_obs          ! Dimension of observed state
  REAL, INTENT(in)    :: state(dim)             ! PE-local model state
  REAL, INTENT(inout) :: m_state(dim_obs) ! Observed state
```

# Building the Assimilation System

Problem reduces to:

1. Insert assimilation subroutine calls to model codes
2. Configuration of parallelization
   (MPI communicators)
3. Implementation of compartment-specific user routines
   and linking with model codes at compile time

# 2-level Parallelism

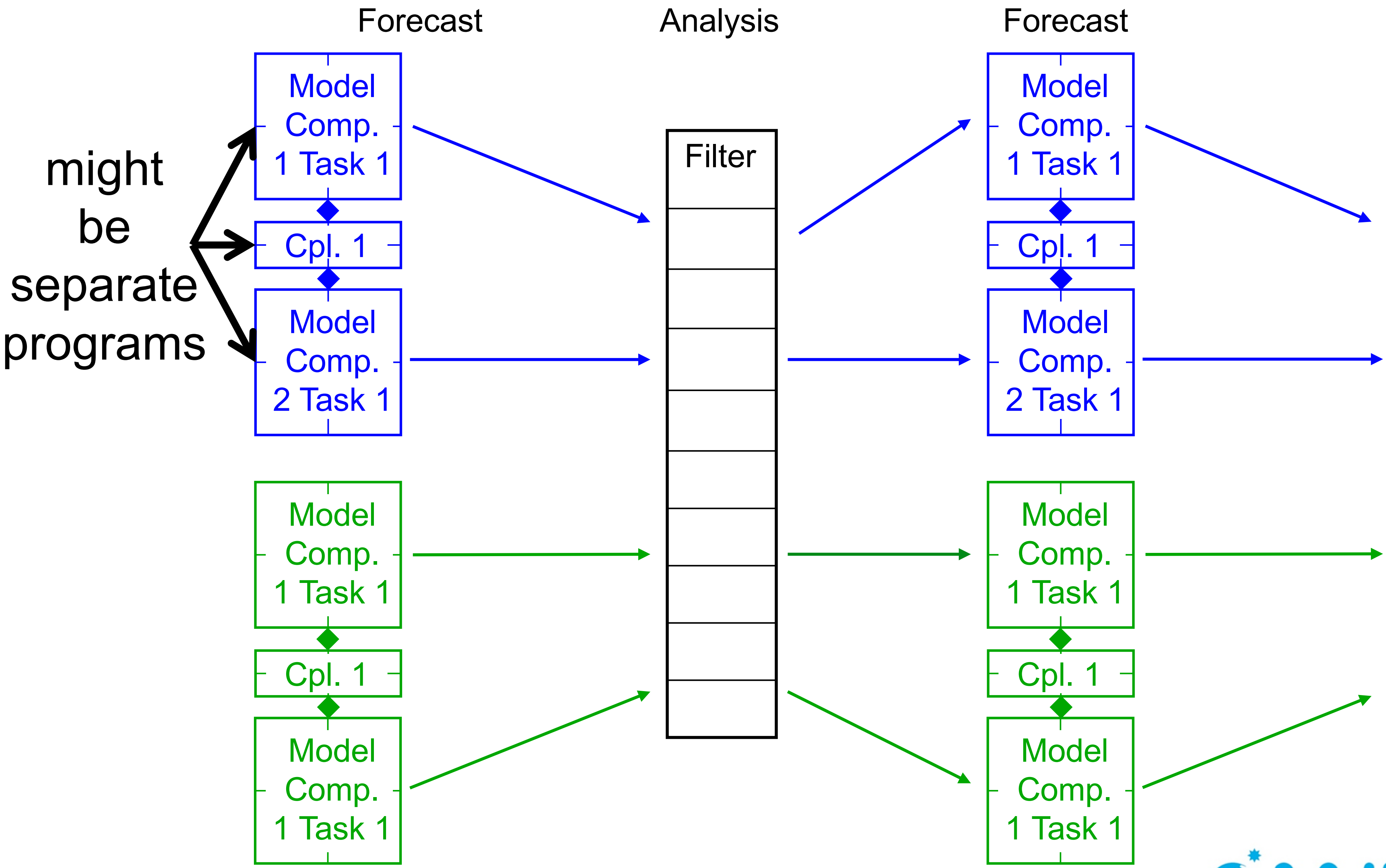



1. Multiple concurrent model tasks
2. Each model task can be parallelized

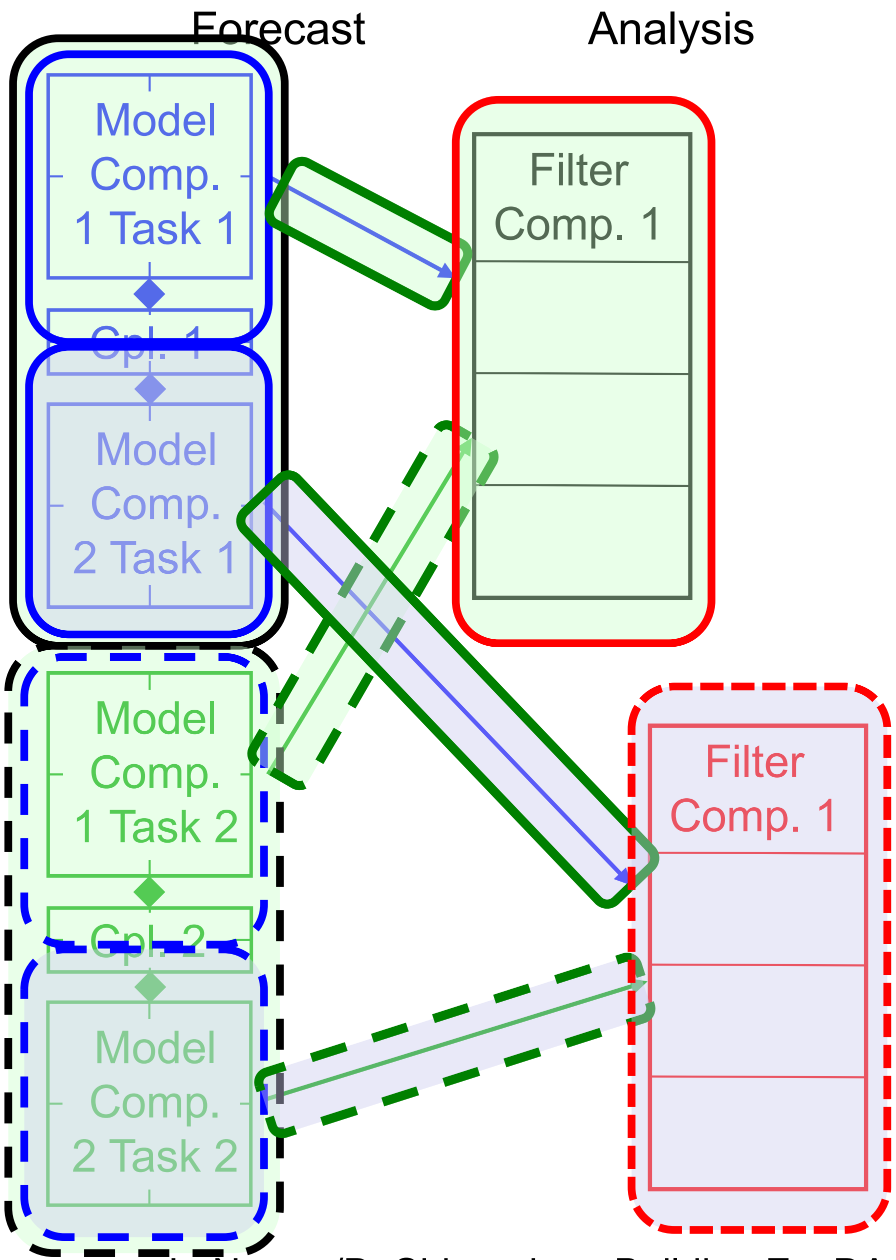
- Analysis step is also parallelized

- Configured by "*MPI Communicators*"

# 2 compartment system – strongly coupled DA



# Configure Parallelization – weakly coupled DA

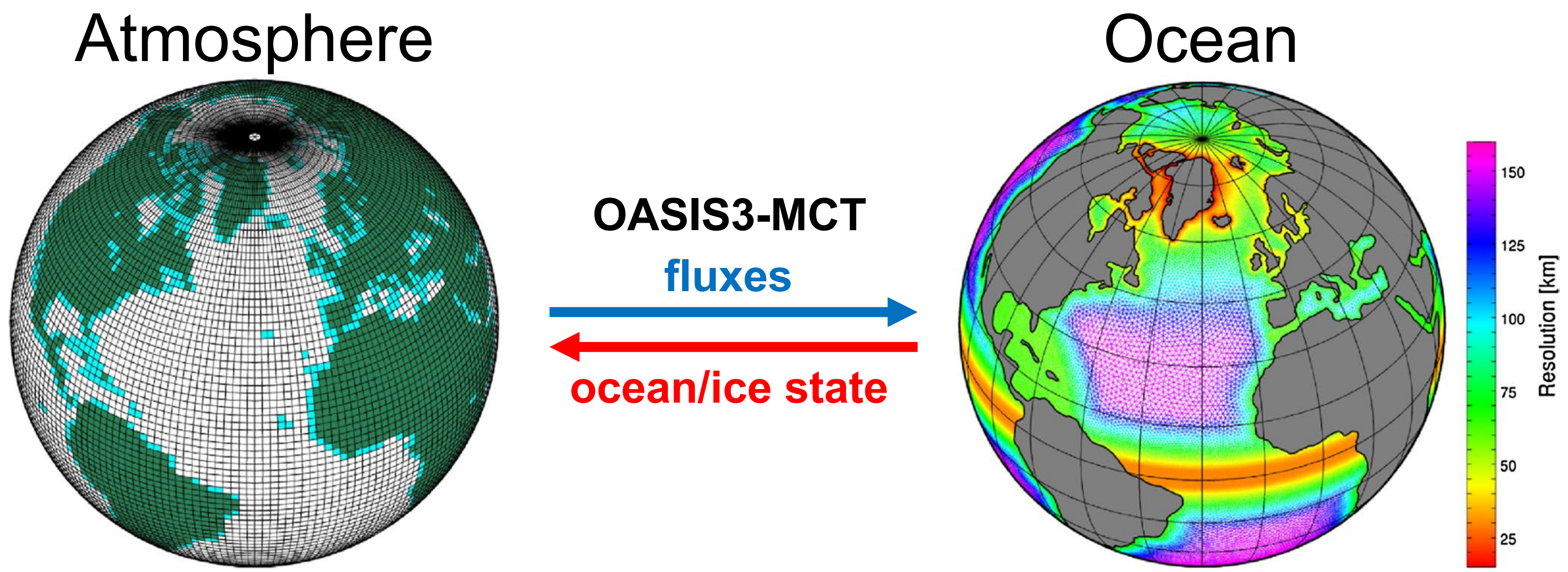



Logical decomposition:

- Communicator for each
  - Coupled model task
  - Compartment in each task (init by coupler)
  - (Coupler *might want to split MPI_COMM_WORLD*)
  - Filter for each compartment
  - Connection for collecting ensembles for filtering

- Different compartments
  - Initialize distinct assimilation parameters
  - Use distinct user routines

# Example: ECHAM6-FESOM



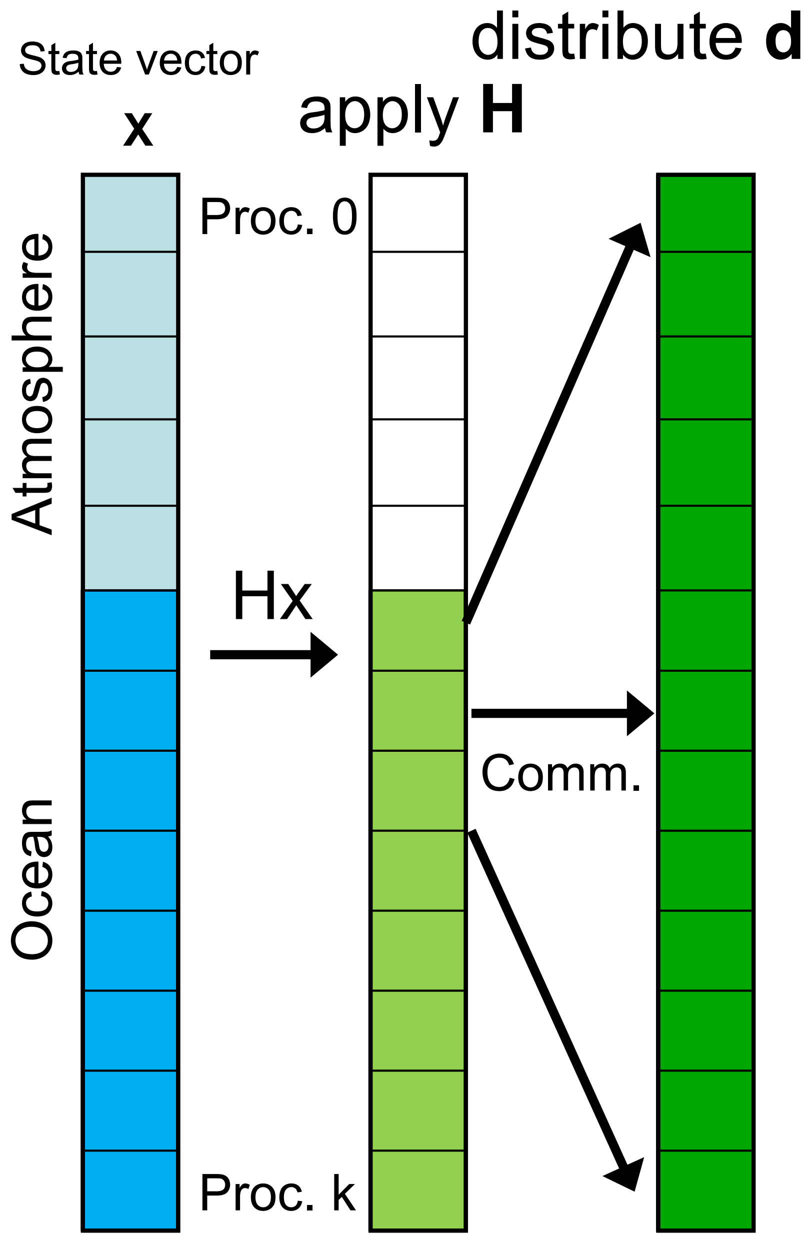

2 executables ECHAM and FESOM – do all coding twice

- add subroutine call into both models
- adapt model communicator (distinct names in the models)
- replace MPI_COMM_WORLD in communication routines for fluxes

In OASIS-MCT library

- Replace MPI_COMM_WORLD in OASIS coupler
- Let each model task write files with interpolation information

# Strongly coupled: Parallelization of analysis step




We need innovation: $\mathbf{d} = \mathbf{Hx} - \mathbf{y}$

Observation operator links different compartments

1. Compute part of **d** on process ‘owning’ the observation

2. Communicate **d** to processes for which observation is within localization radius

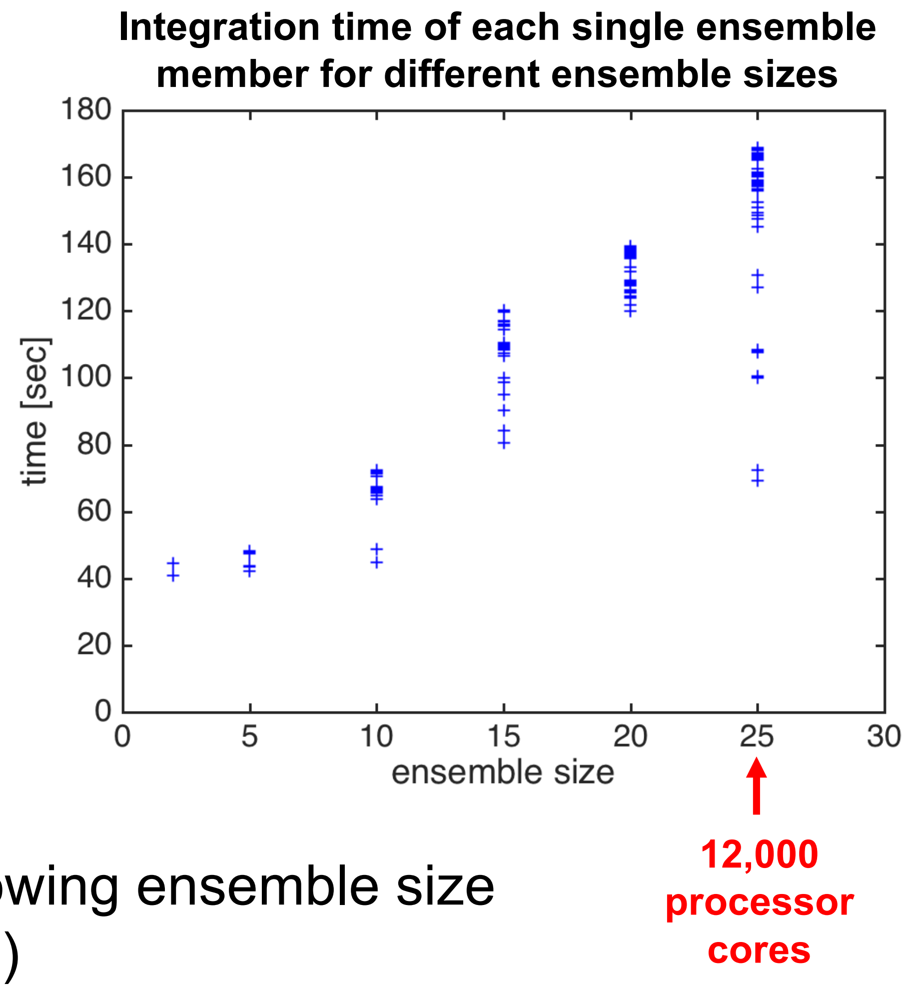
# Execution times (weakly-coupled, DA only into ocean)

MPI-tasks

- ECHAM: 288
- FESOM: 192

Timings (1 day):

- Ens. forecast: 40 – 168 sec
- Analysis step: 0.5 – 0.9 sec




A remaining issue:

- Increasing integration time with growing ensemble size (Factor 4 for 12-fold ensemble size)
- Large variability in integration time over ensemble tasks
- Likely caused by MPI-communication (e.g. no optimal distribution of programs over compute nodes/racks)

# Summary

- Status of AWI-CM/PDAF: ready to be used (Postdoc just started)
- Software framework simplifies building data assimilation systems
- Efficient online DA coupling; minimal model code changes
- Setup of data assimilation with coupled model
  1. Configuration of communicators
  2. Add routines for initialization & analysis step
  3. Implementation of case-specific user-routines
- Size of computing problem and communication layout might lead to tuning requirements

Thank you !

# Current algorithms in PDAF

PDAF Parallel Data Assimilation Framework

PDAF originated from comparison studies of different filters

**Filters and smoothers**

- EnKF (Evensen, 1994 + perturbed obs.)
- ETKF (Bishop et al., 2001)
- SEIK filter (Pham et al., 1998)
- ESTKF (Nerger et al., 2012)
- NETF (Toedter & Ahrens, 2015)

Not yet released:
- serial EnSRF
- particle filter
- EWPF

**All methods include**

- global and localized versions
- smoothers



AWI

# References


- http://pdaf.awi.de

- Nerger, L., Hiller, W. *Software for Ensemble-based DA Systems – Implementation and Scalability*. Computers and Geosciences 55 (2013) 110-118

- Nerger, L., Hiller, W., Schröter, J.(2005). *PDAF - The Parallel Data Assimilation Framework: Experiences with Kalman Filtering*, Proceedings of the Eleventh ECMWF Workshop on the Use of High Performance Computing in Meteorology, Reading, UK, 25 - 29 October 2004, pp. 63-83.