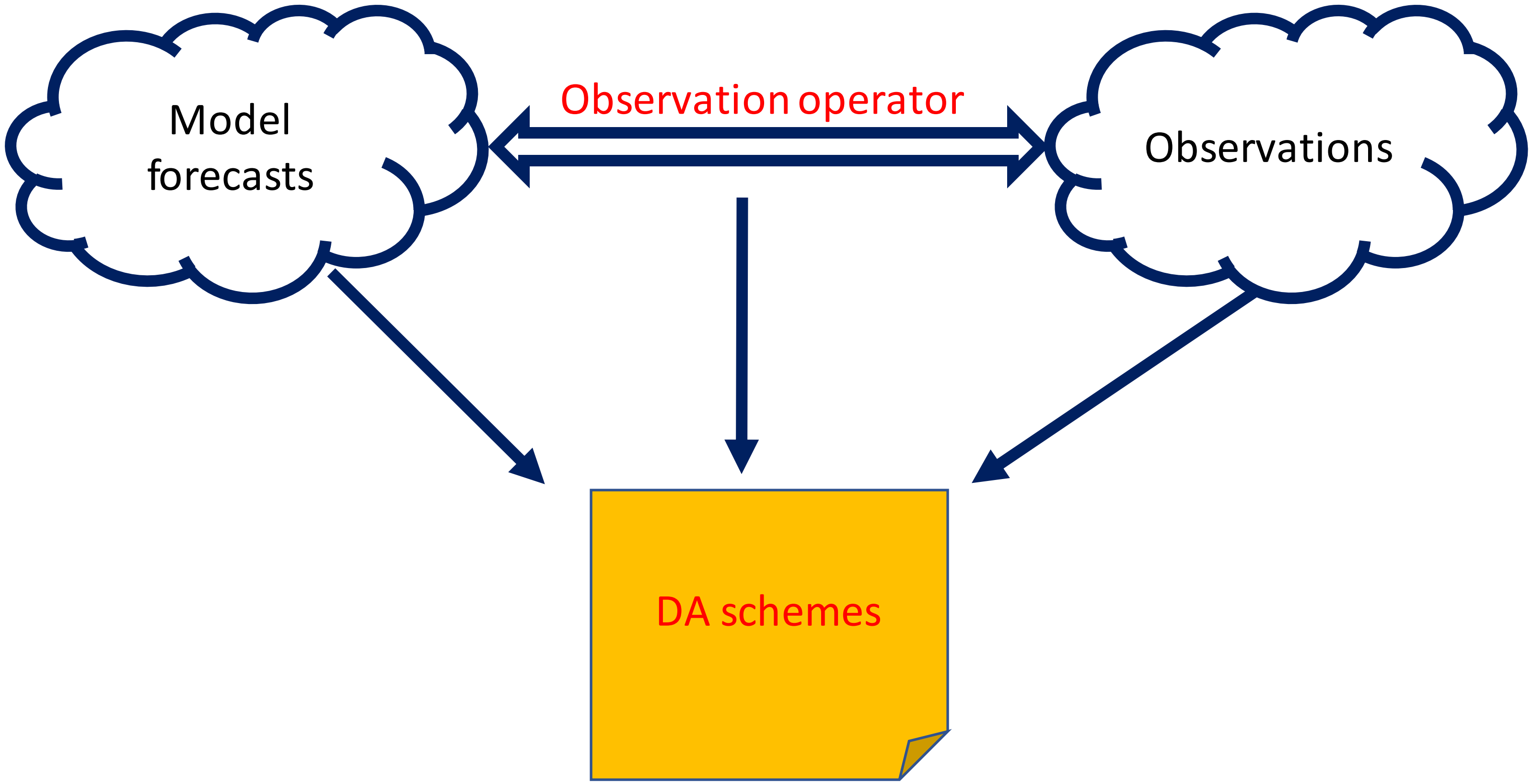
Data assimilation software

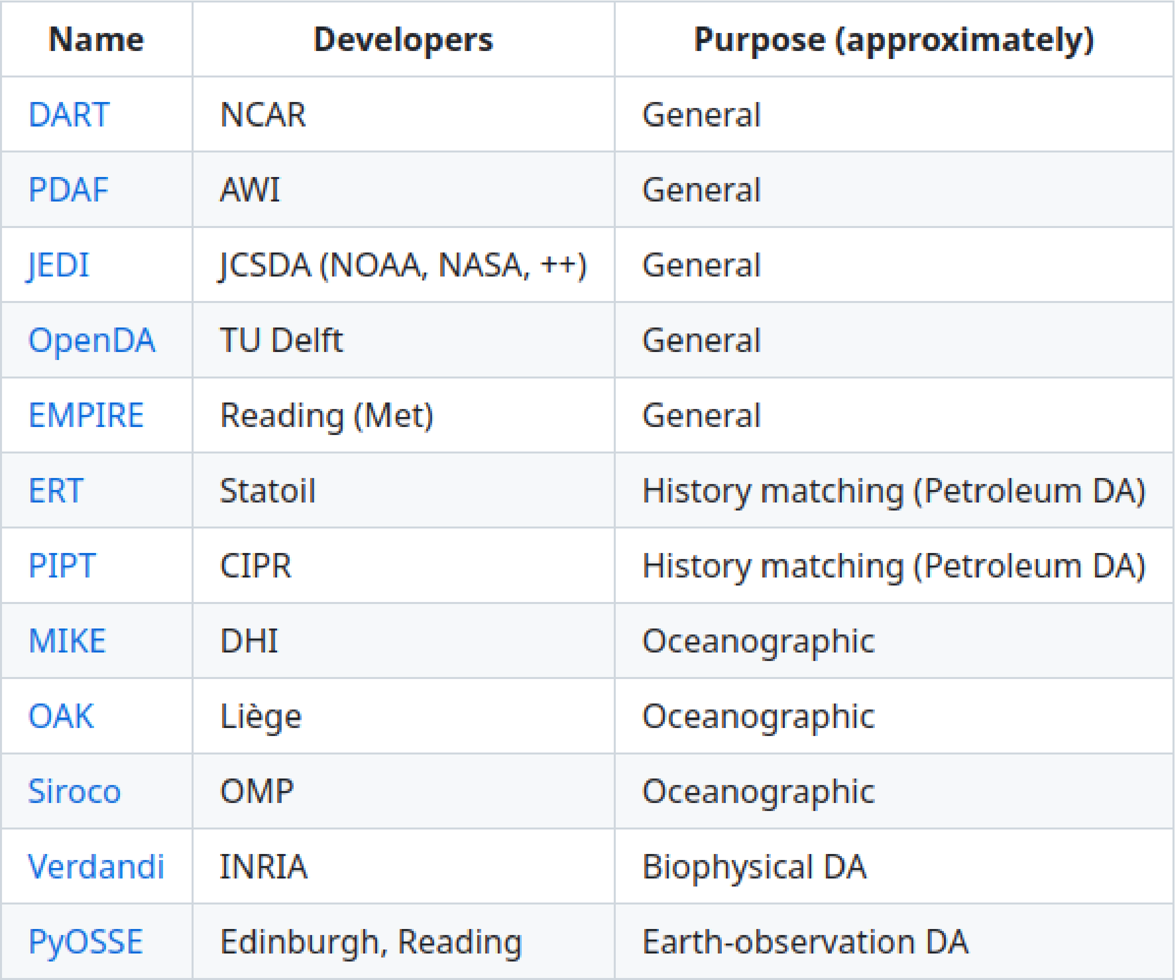
Yumeng Chen



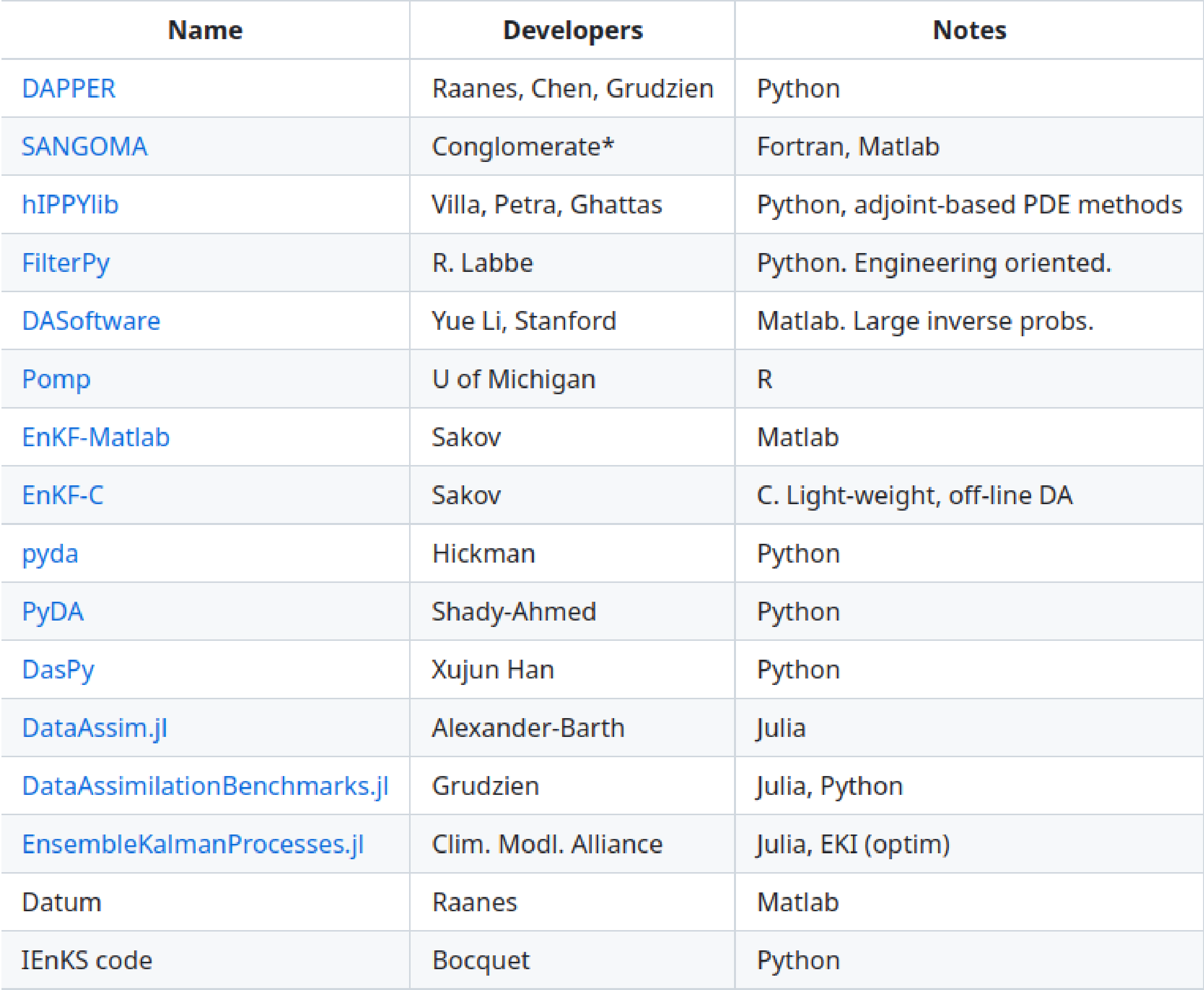
- Efficient DA algorithm implementation is time consuming
- Avoiding of writing and debugging code
- Focus on the scientific questions
- Ensures reproducible and consistent scientific research

A quick recap and outlook

- Most popular DA methods used in weather and climate are variational methods and (Ensemble) Kalman filters (EnKF)
- Both methods use background and observation error covariances
- In the EnKF, the background error covariance is estimated from an ensemble of model forecasts
- You will see more about EnKF tomorrow



Name	Developers	Purpose (approximately)
DART	NCAR	General
PDAF	AWI	General
JEDI	JCSDA (NOAA, NASA, ++)	General
OpenDA	TU Delft	General
EMPIRE	Reading (Met)	General
ERT	Statoil	History matching (Petroleum DA)
PIPT	CIPR	History matching (Petroleum DA)
MIKE	DHI	Oceanographic
OAK	Liège	Oceanographic
Siroco	OMP	Oceanographic
Verdandi	INRIA	Biophysical DA
PyOSSE	Edinburgh, Reading	Earth-observation DA



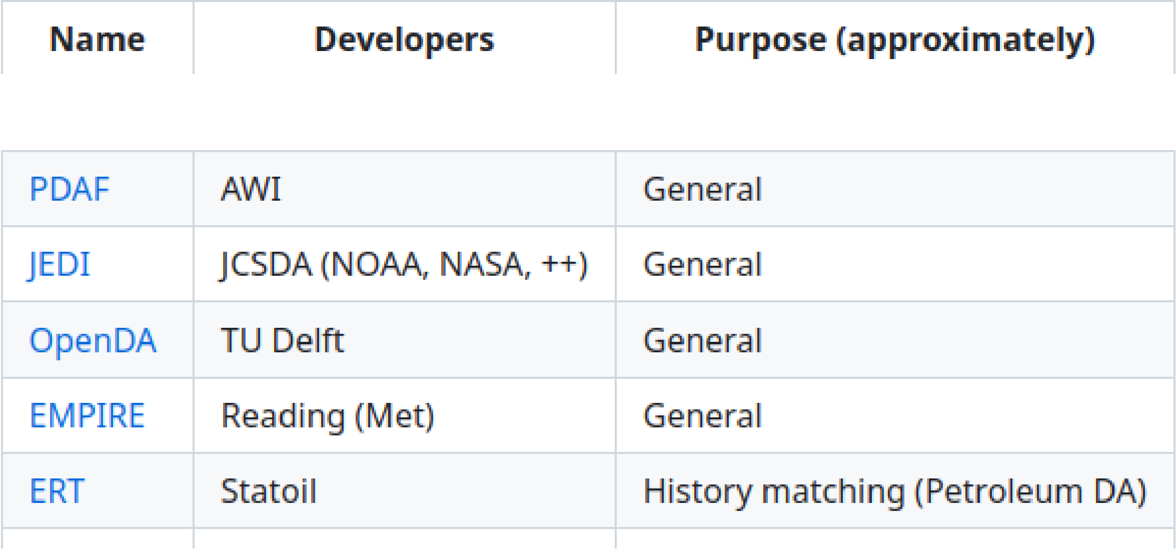
Name	Developers	Notes
DAPPER	Raanes, Chen, Grudzien	Python
SANGOMA	Conglomerate*	Fortran, Matlab
hIPPYlib	Villa, Petra, Ghattas	Python, adjoint-based PDE methods
FilterPy	R. Labbe	Python. Engineering oriented.
DASoftware	Yue Li, Stanford	Matlab. Large inverse probs.
Pomp	U of Michigan	R
EnKF-Matlab	Sakov	Matlab
EnKF-C	Sakov	C. Light-weight, off-line DA
pyda	Hickman	Python
PyDA	Shady-Ahmed	Python
DasPy	Xujun Han	Python
DataAssim.jl	Alexander-Barth	Julia
DataAssimilationBenchmarks.jl	Grudzien	Julia, Python
EnsembleKalmanProcesses.jl	Clim. Modl. Alliance	Julia, EKI (optim)
Datum	Raanes	Matlab
IEnKS code	Bocquet	Python

Which one should I choose?

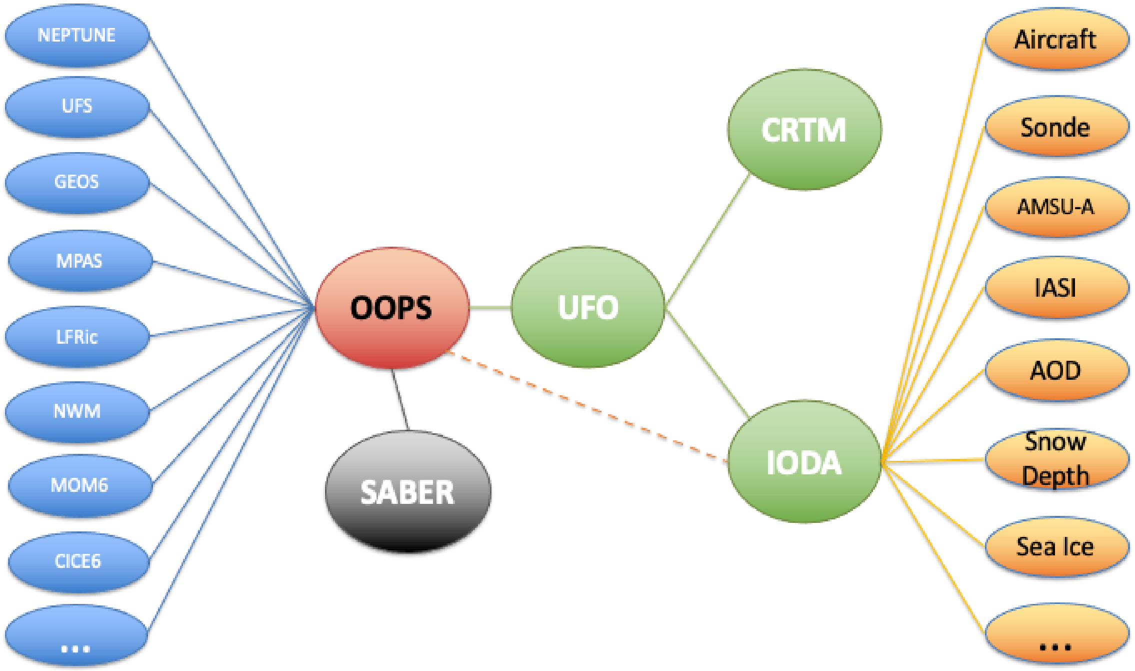
Operational use/research for large models

Methodology research for small models like Lorenz 96

Courtesy:
Patrick Raanes on https://github.com/nansencenter/DAPPER/blob/master/README.md



Name	Developers	Purpose (approximately)
PDAF	AWI	General
JEDI	JCSDA (NOAA, NASA, ++)	General
OpenDA	TU Delft	General
EMPIRE	Reading (Met)	General
ERT	Statoil	History matching (Petroleum DA)

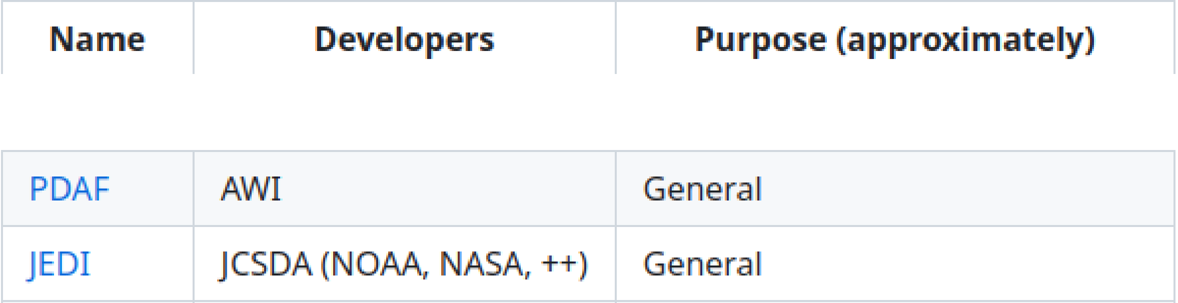


Courtesy: JEDI documentation

- Various operational centres opt for Joint Effort for Data assimilation Integration (JEDI) developed by JCSDA, including UKMO, NOAA, etc.

- JEDI consists of a few components:
    - OOPS: Object Oriented Prediction System:
        - 3D-Var; 4DEnsVar; 4DVar; Weak constraint 4DVar
        - LETKF, LGETKF – two types of EnKF
    - SABER: System Agnostic Background Error Representation
        - computing and working with the background error covariance matrix
    - IODA: Interface for Observation Data Access
        - handle an immense amount of data from the providers
    - UFO: Unified Forward Operator
        - Obs. Operator

- Diverse functionalities, suitable for operational weather and climate models



Name	Developers	Purpose (approximately)
PDAF	AWI	General
JEDI	JCSDA (NOAA, NASA, ++)	General

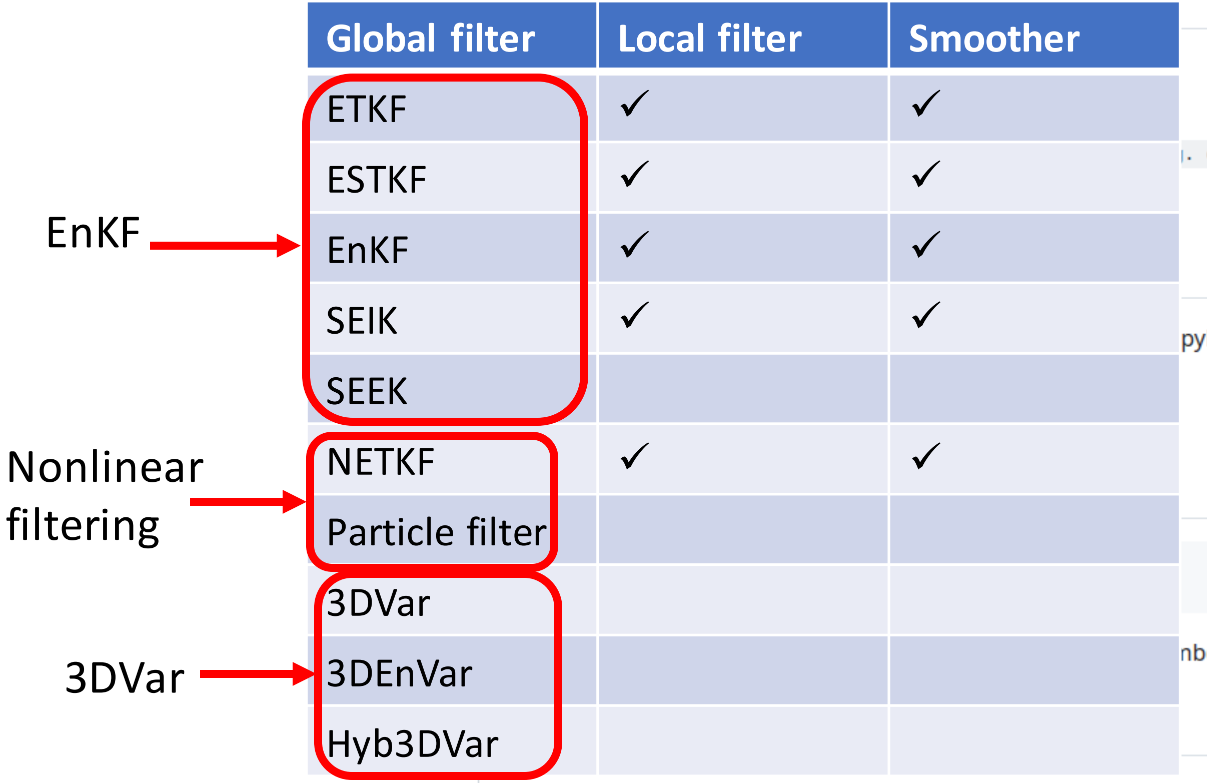
PDAF Parallel Data Assimilation Framework

A Python interface to the Fortran-written data assimilation library - PDAF

test_build passing

Prerequisite:

https://github.com/yumengch/pyPDAF



Global filter	Local filter	Smoother
ETKF	✓	✓
ESTKF	✓	✓
EnKF	✓	✓
SEIK	✓	✓
SEEK		
NETKF	✓	✓
Particle filter		
3DVar		
3DEnVar		
Hyb3DVar		

EnKF
Nonlinear filtering
3DVar

. openMPI/MPICH

pyPDAF. Hence, the Fortran compiler options

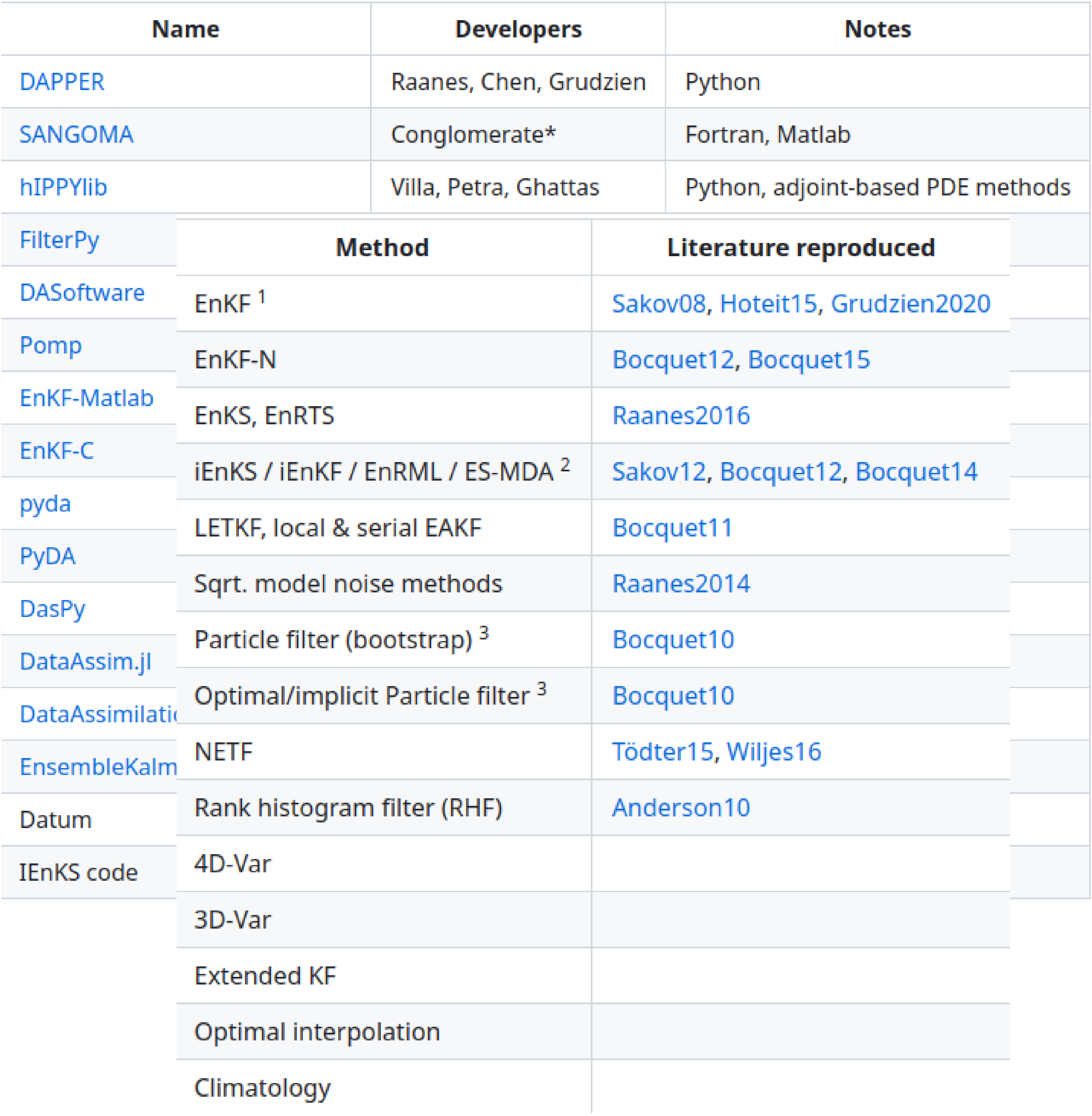
nber uses 1 process.

Currently, it interfaces with subroutines of `PDAF-V2.0` with an example for online coupling with PDAF using a simple model based on the tutorial from PDAF. Some interface in Python changes slightly due to different ways to handling return values in Python from Fortran.

- Parallel Data Assimilation Framework (PDAF) by Lars Nerger in Alfred Wegener Institute
- Simple observation handling, ensemble generation (background error), diagnostics routines
- The communication between the model, observations and PDAF is done through well-defined user interface in Fortran – PDAF is more of a library
- Suitable for weather and climate models, e.g. AWI-CM, MITgcm, MPI-ESM, NEMO, etc.
- Focus on ensemble Kalman filter (EnKF)
- There is a Python interface to the PDAF
- We will come back talk about PDAF later

DAPPER

- *DAPPER is developed mainly by Patrick Raanes*
- *DAPPER is a set of templates for DA methods*
- *The typical set-up is a synthetic (twin) experiment as what we do in practicals*
- *Ease of adding new DA methods and models*
- *Purely in Python, suitable for methodology research and development using small models such as Lorenz models*
- *It was quite helpful when I learn DA*



Name	Developers	Notes
DAPPER	Raanes, Chen, Grudzien	Python
SANGOMA	Conglomerate*	Fortran, Matlab
hIPPYlib	Villa, Petra, Ghattas	Python, adjoint-based PDE methods
FilterPy		
DASoftware		
Pomp		
EnKF-Matlab		
EnKF-C		
pyda		
PyDA		
DasPy		
DataAssim.jl		
DataAssimilati		
EnsembleKalm		
Datum		
IEnKS code		

Method	Literature reproduced
EnKF [1]	Sakov08, Hoteit15, Grudzien2020
EnKF-N	Bocquet12, Bocquet15
EnKS, EnRTS	Raanes2016
iEnKS / iEnKF / EnRML / ES-MDA [2]	Sakov12, Bocquet12, Bocquet14
LETKF, local & serial EAKF	Bocquet11
Sqrt. model noise methods	Raanes2014
Particle filter (bootstrap) [3]	Bocquet10
Optimal/implicit Particle filter [3]	Bocquet10
NETF	Tödter15, Wiljes16
Rank histogram filter (RHF)	Anderson10
4D-Var	
3D-Var	
Extended KF	
Optimal interpolation	
Climatology	

DAPPER

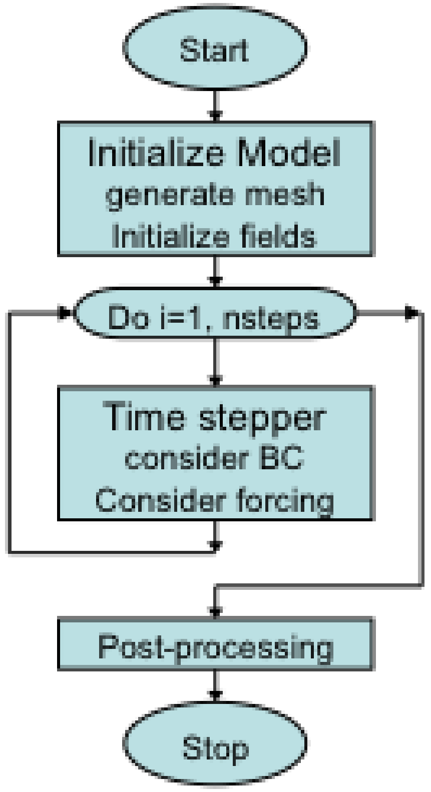
https://shorturl.at/atuzA

https://shorturl.at/dnNUX

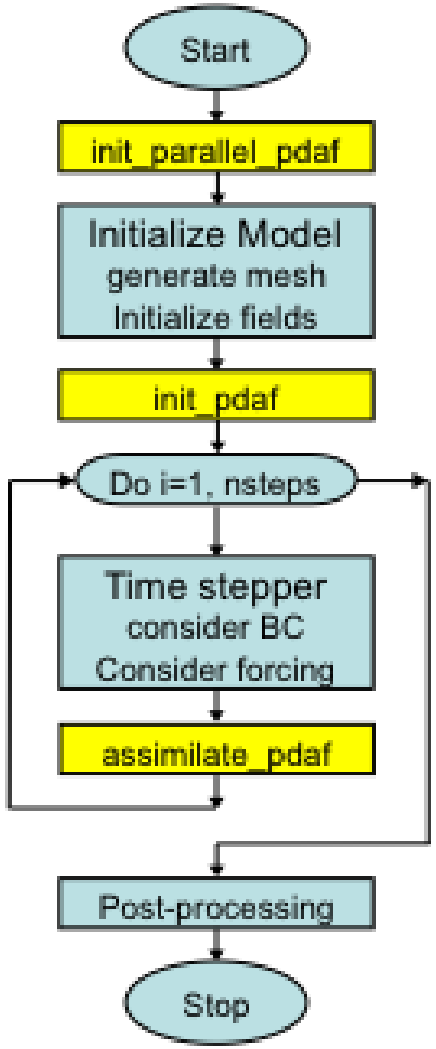
https://github.com/nansencenter/DAPPER

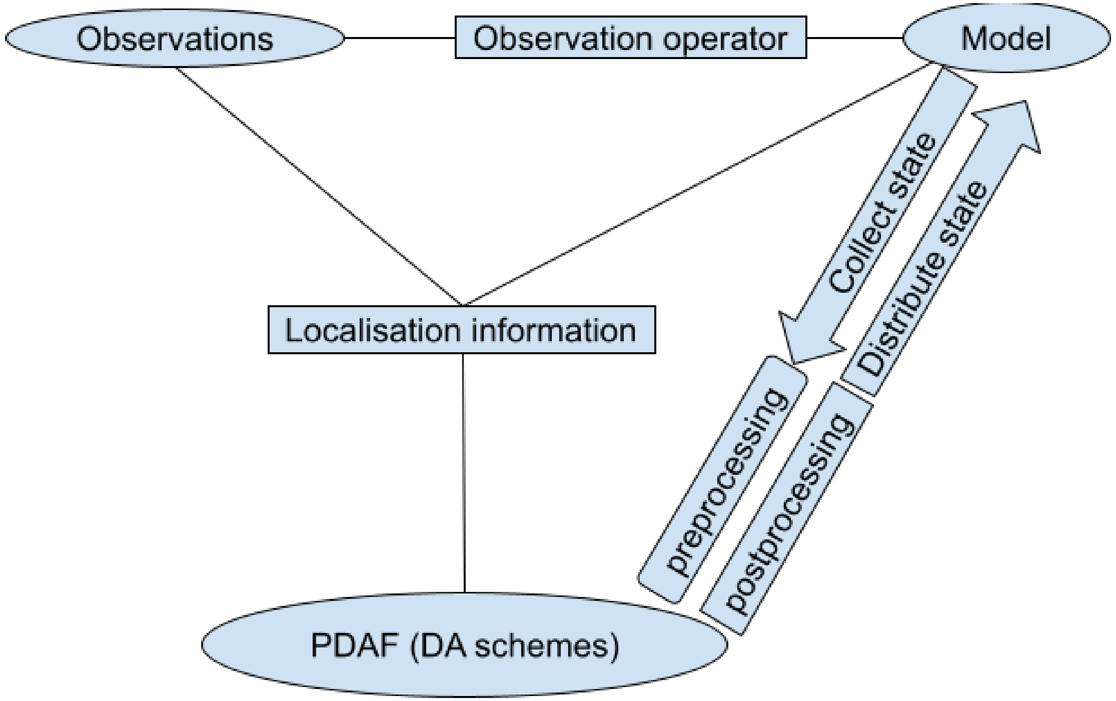
General idea of PDAF implementations

Simulation Model



'fully-parallel' assimilation system



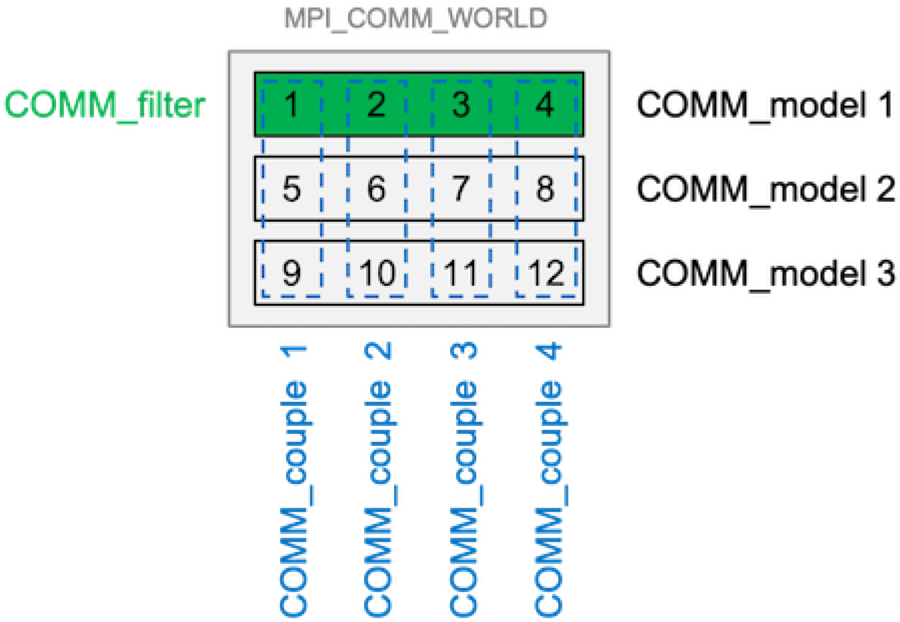


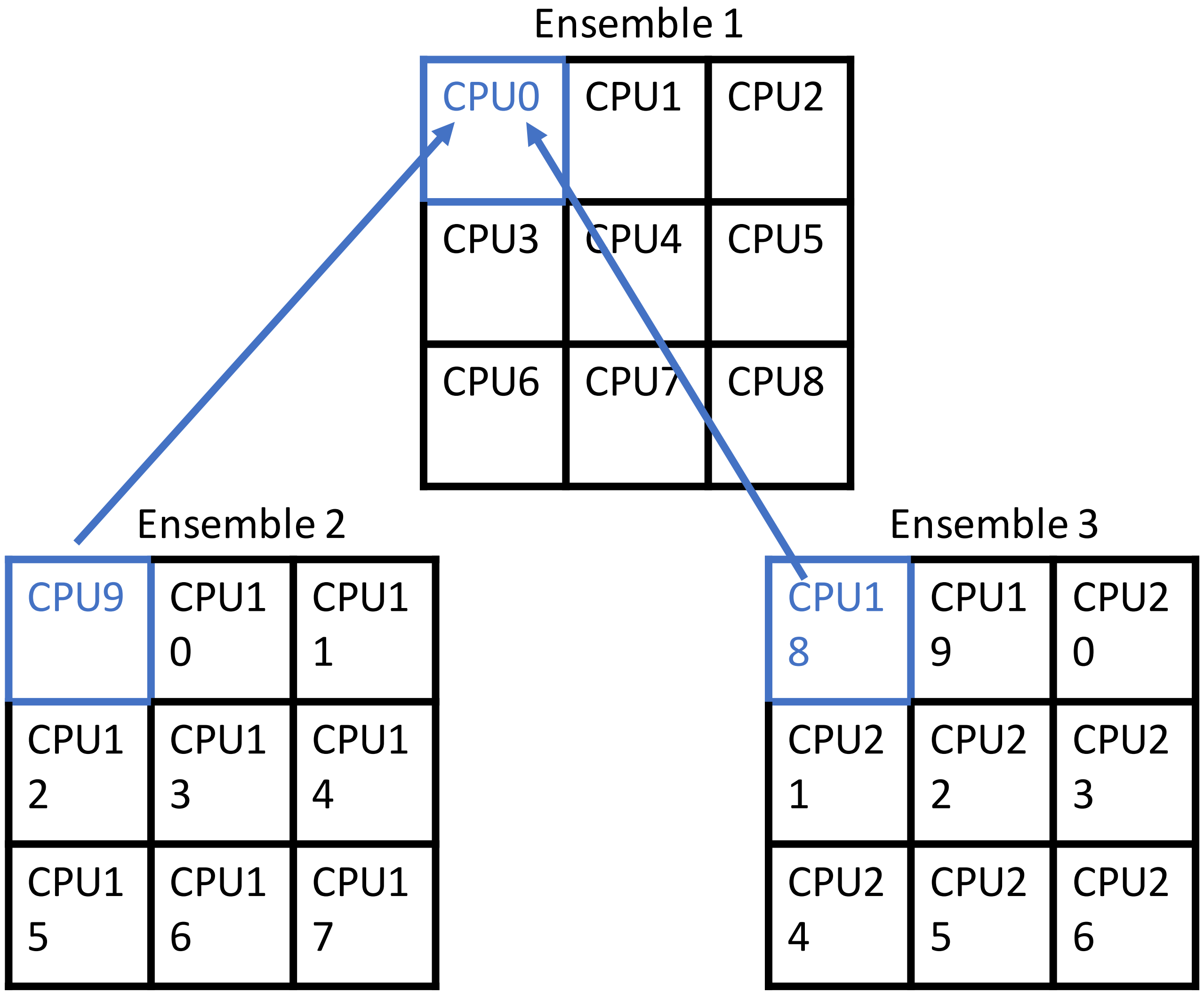
Online mode:

PDAF is attached to the model source code by minimal changes to the code

The assimilation runs together with the model integration

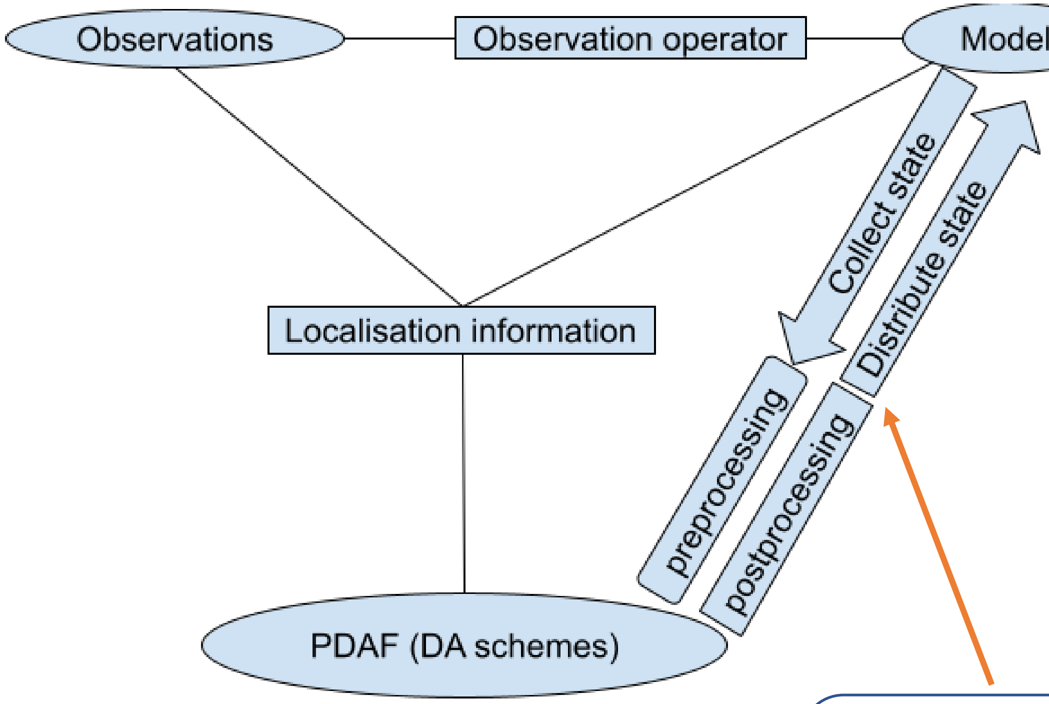
PDAF Parallel
Data Assimilation
Framework



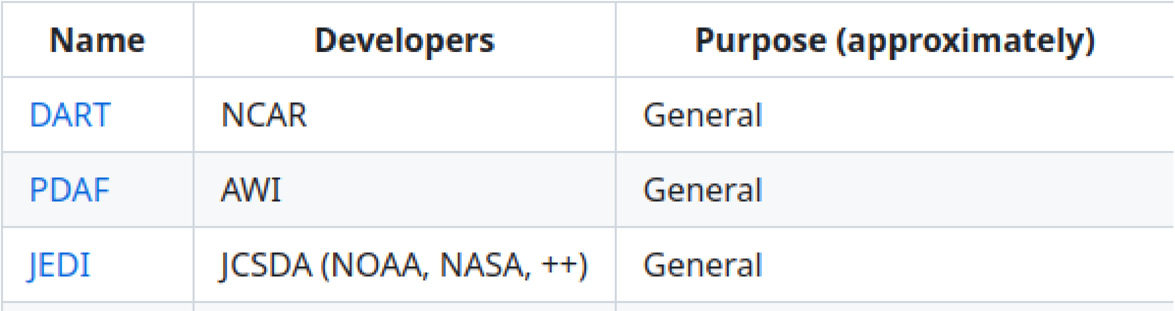


- Weather and climate models run on multiple processors
- Each processor calculates for a fraction of model domain
- Information on each processors can exchange with each other
- PDAF make uses of this feature

- Offline mode
  - PDAF is a separate program from the model.
  - The model information is read from the model output/restart files



Read from/write to data files



Name	Developers	Purpose (approximately)
DART	NCAR	General
PDAF	AWI	General
JEDI	JCSDA (NOAA, NASA, ++)	General

- DA software can avoid us coding the sometimes complicated algorithms

- Some common DA software for large models: JEDI/PDAF

- Each DA software/library has its own features and limitations
    - Choosing the correct DA software can depend on various factors, e.g., the applications, the DA method, the model complexity

EAT: Ensemble and Assimilation Tool

- Model: One-dimensional ocean model GOTM
  - describes physical processes in marine and freshwater water columns.

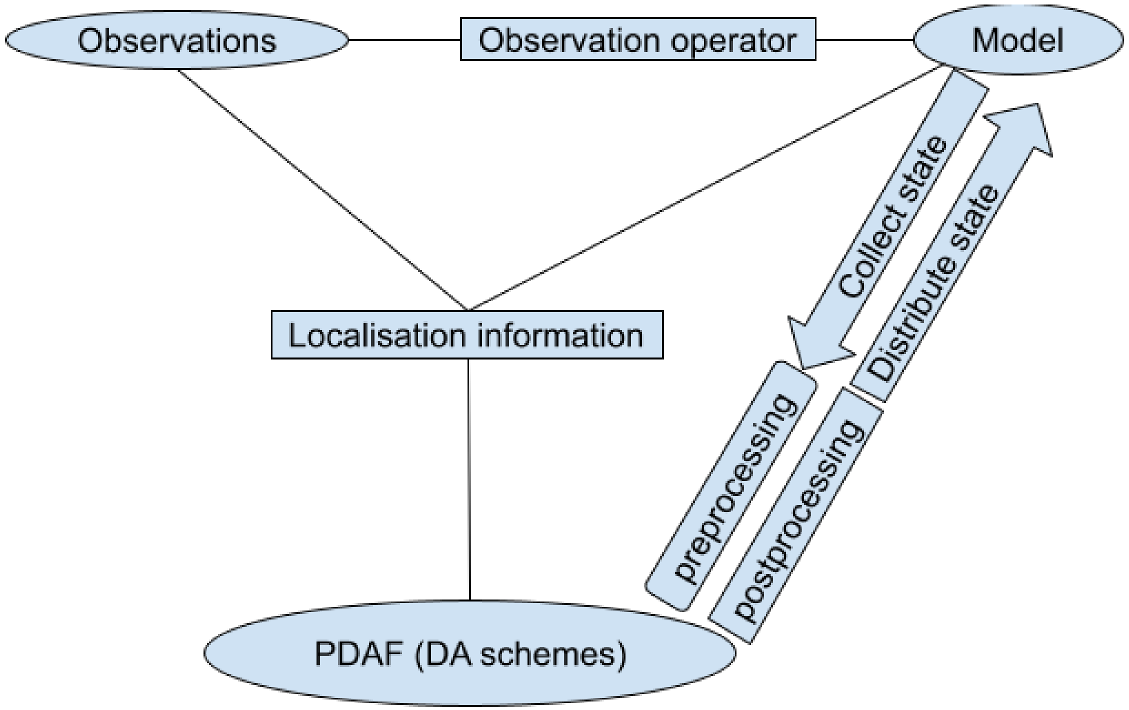
- DA: PDAF using Python interface

- Example: the Northern North Sea.
  - Assimilate sea surface temperature observations from the SST CCI.
  - https://shorturl.at/ftxA7

BoldingBruggeman/eat: Ensemble and Assimilation Tool (EAT)
(github.com)

ImplementAnalysisLocal – PDAF - Parallel Data Assimilation Framework (awi.de)

1. Collect state vector from model fields (U_collect_state)
2. Preprocess the ensemble (U_prepoststep)
3. provides the number of local analysis domains (U_init_n_domains)
4. initializes the observation information and provides the size of observation vector (U_init_dim_obs_pdafomi)
5. Apply the observation operator (U_obs_op_pdafomi)
6. provides the state dimension for a local analysis domain (U_init_dim_l)
7. initializes the size of the observation vector for a local analysis domain (U_init_dim_obs_l_pdafomi)
8. initializes a local state vector from the global state vector (U_g2l_state)
9. initializes the corresponding part of the global state vector from the provided local state vector (U_l2g_state)
10. Actual assimilation
11. Post-process the ensemble (U_prepoststep)
12. Distribute analysis ensemble to model fields (U_distribute_state)
13. determines the next assimilation step (U_next_observation)


Observations
Observation operator
Model
Collect state
Distribute state
Localisation information
preprocessing
postprocessing
PDAF (DA schemes)