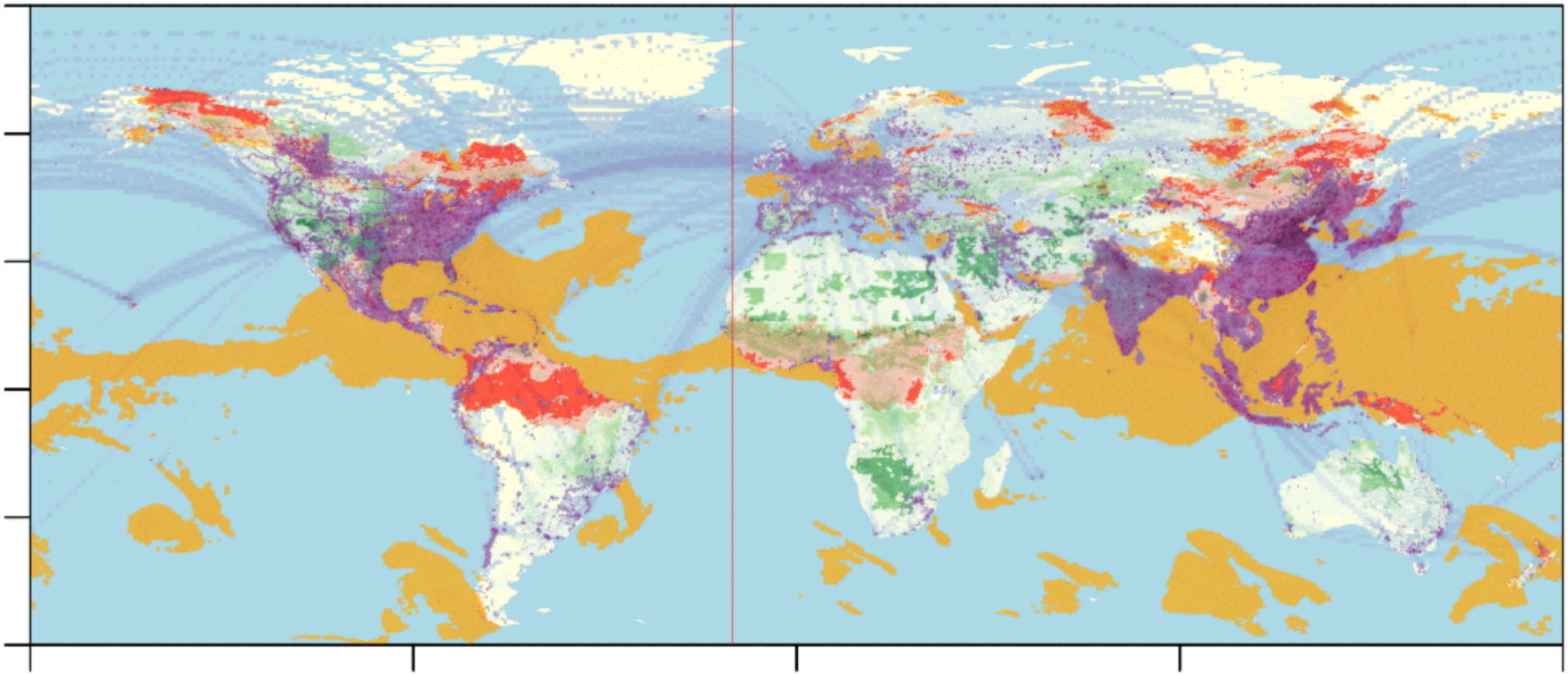


Current HEMCO Structure and Capabilities

GEOS-Chem NCAR Visit
Melissa Sulprizio (Harvard)
January 8, 2020

Image from Christoph Keller

HEMCO overview

- HEMCO was originally developed by Christoph Keller to streamline emissions processing in GEOS-Chem

Before HEMCO	With HEMCO
Each emissions inventory required its own Fortran module to read in data, perform unit conversions, and apply scale factors.	All emissions input, scale factors, and masks are listed in a configuration file. Emissions are automatically regridded to the model grid, scale factors/masks are applied, and inventories are combined following a specified hierarchy.
Emissions data needed to be preprocessed to the model grid (4°x5° global, 2°x2.5° global, or 0.5°x0.666° regional).	Emissions data can be read in at any resolution, preferably the highest resolution available.
Diagnostics were hardwired in GEOS-Chem for each species/inventory and saved out on the model grid to BPCH format.	Diagnostics are specified at run time via an input file and are saved out on the model grid to netCDF format.

HEMCO overview continued

- HEMCO can also compute emissions that vary based on external fields (meteorology, species concentrations) via “Extensions”
  - The external fields can be provided by an external model or read into HEMCO directly

- GEOS-Chem now uses HEMCO for reading in all geospatial input data, not just emissions
  - HEMCO now also reads meteorology, chemistry input data, restart files, boundary conditions
  - This has the advantage of consistently reading, regridding, cropping, and scaling input data

HEMCO capabilities

File formats	NetCDF (COARDS or CF compliant) NOTE: Some extensions also read point sources or look-up tables in ASCII format
Unit conversions	Fluxes converted automatically to kg/m2/s
Horizontal regridding	MAP_A2A
Vertical regridding	MESSy
Time cycling	Cycle, Range, Exact, Average, Interpolate (more options added as needed)
Masking	Lat/lon ranges or gridded fields supported
Scale factors	Scalar values or gridded fields supported (monthly, hourly, daily automatically recognized)

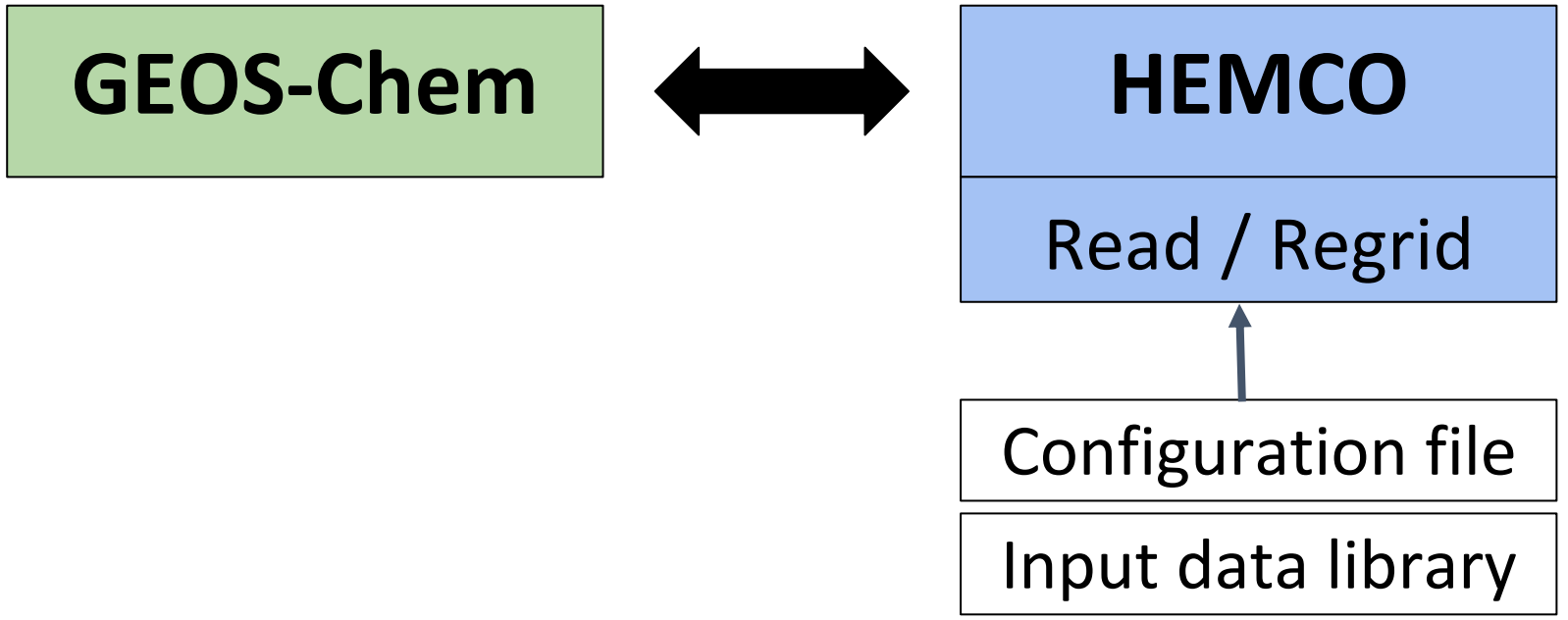
Connecting HEMCO to other models

- External models are connected to HEMCO via an interface module
  - Current interfaces: HEMCO Standalone, GEOS-Chem, and ESMF (for GCHP, GEOS)
  - Interfaces pass information from the model to HEMCO, run HEMCO to compute emissions, then pass fluxes from HEMCO to the model
  - HEMCO can easily be connected to other models by creating a new interface

- HEMCO is currently distributed with the GEOS-Chem source code but is in the process being split off into its own repository on Github
  - See: https://github.com/geoschem/HEMCO
  - HEMCO will be a Git submodule in GEOS-Chem 13.0.0 (Spring 2020)
  - This will also allow for HEMCO to be easily dropped into other models

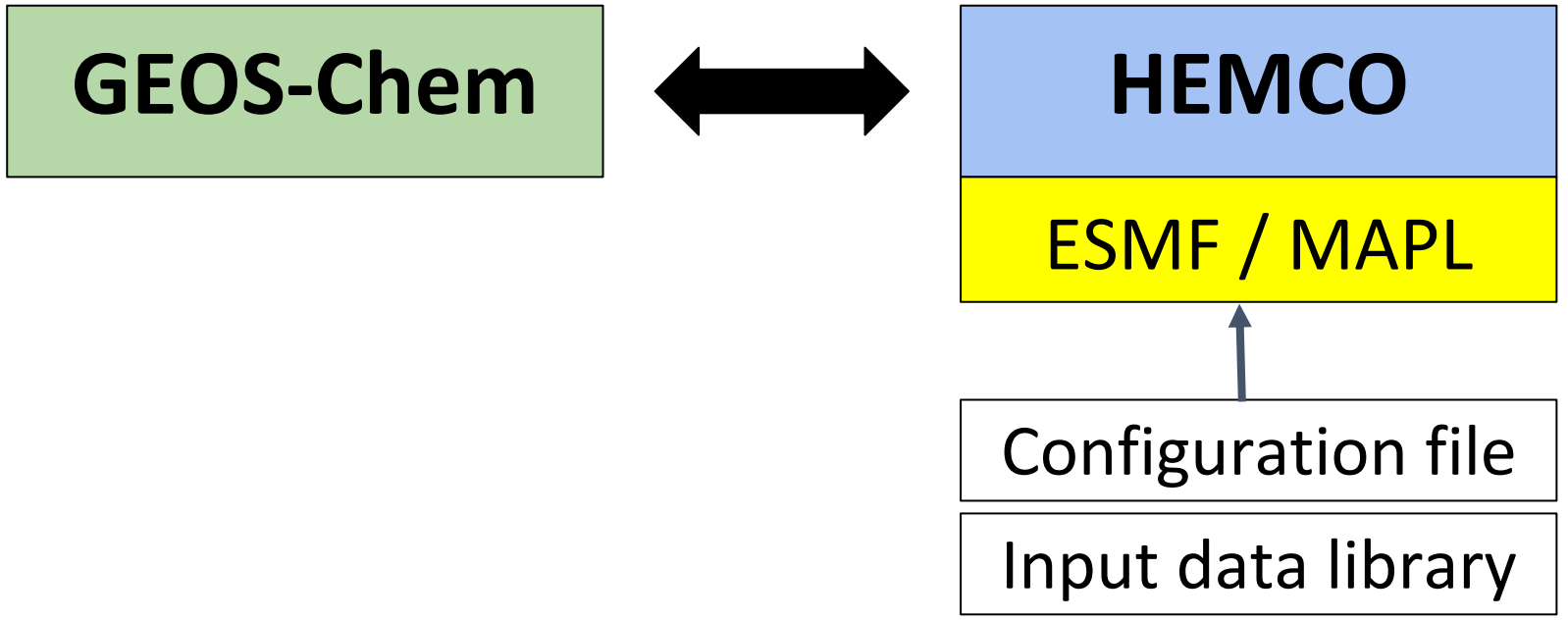
HEMCO in GEOS-Chem vs GCHP/GEOS

	GEOS-Chem Classic	GCHP / GEOS
I/O and regridding handled by	HEMCO	MAPL / ESMF
Grids accepted	Latitude-longitude only	Rectilinear, curvilinear, or unstructured
Input data settings	HEMCO_Config.rc	ExtData.rc + HEMCO_Config.rc
Diagnostic settings	HEMCO_Diagn.rc	HISTORY.rc + HEMCO_Diagn.rc

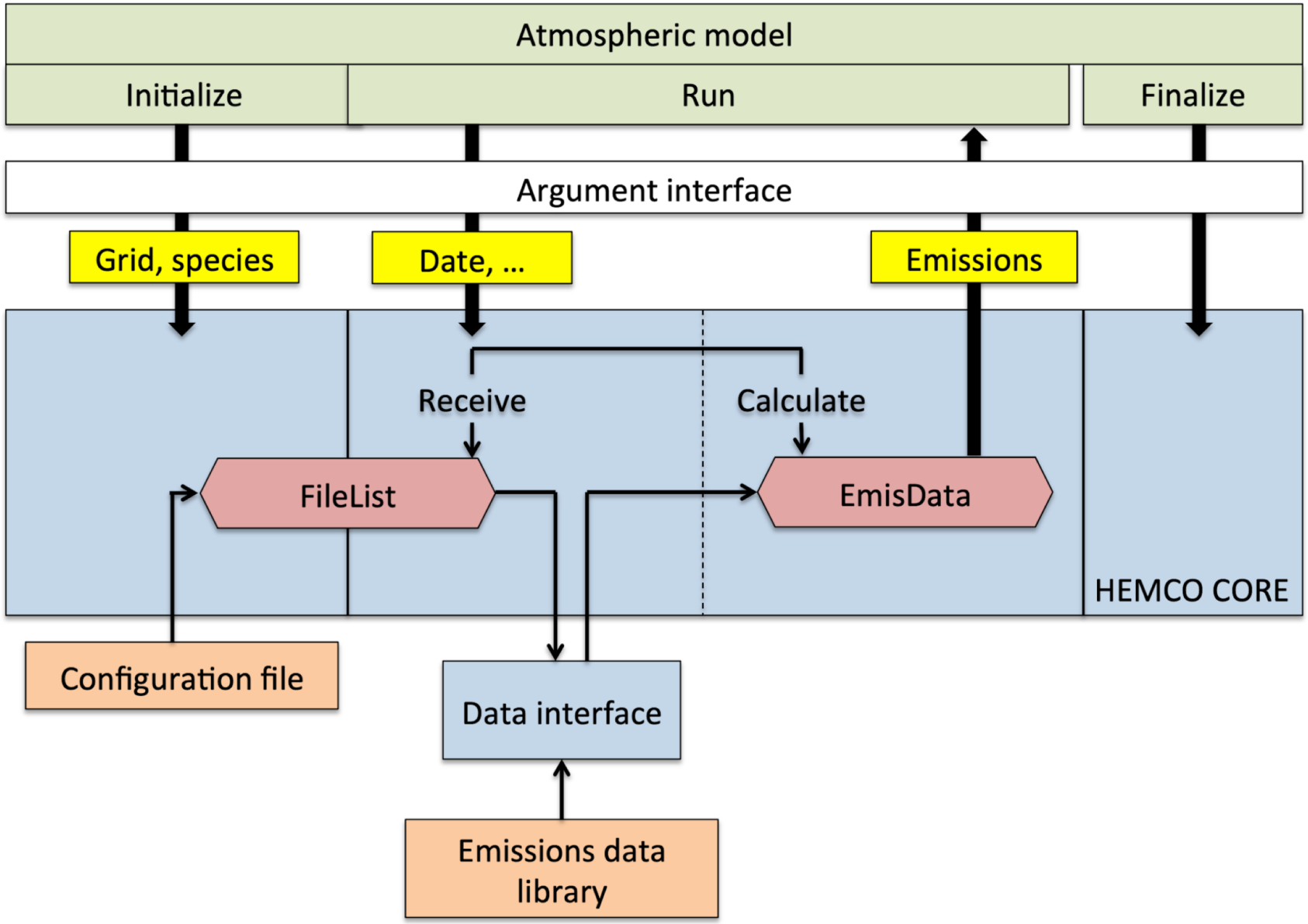


HEMCO in GEOS-Chem vs GCHP/GEOS

	GEOS-Chem Classic	GCHP / GEOS
I/O and regridding handled by	HEMCO	MAPL / ESMF
Grids accepted	Latitude-longitude only	Rectilinear, curvilinear, or unstructured
Input data settings	HEMCO_Config.rc	ExtData.rc + HEMCO_Config.rc
Diagnostic settings	HEMCO_Diagn.rc	HISTORY.rc + HEMCO_Diagn.rc



HEMCO phases



Keller et al. (2014)

HEMCO directory structure

```
HEMCO:
CMakeLists.txt  Core/  Extensions/  Interfaces/  Makefile  README.md
```

- Source code is currently split into:
  1. Core
  2. Extensions
  3. Interfaces

- HEMCO can currently be compiled with GNU Make and CMake
  a. Support for GNU Make will be retired in GEOS-Chem 13.0.0 (Spring 2020)

HEMCO/Core

```
Core:
CMakeLists.txt*          hco_fluxarr_mod.F90       hco_scale_mod.F90
hco_arr_mod.F90          hco_geotools_mod.F90      hco_state_mod.F90
hco_calc_mod.F90         hco_interp_mod.F90        hco_tidx_mod.F90
hco_chartools_mod.F90    hcoio_dataread_mod.F90    hco_timeshift_mod.F90
hco_clock_mod.F90        hcoio_diagn_mod.F90       hco_types_mod.F90
hco_config_mod.F90       hcoio_messy_mod.F90       hco_unit_mod.F90
hco_datacont_mod.F90     hcoio_read_esmf_mod.F90   hco_vertgrid_mod.F90
hco_diagn_mod.F90        hcoio_read_std_mod.F90    interpreter.F90
hco_driver_mod.F90       hcoio_write_esmf_mod.F90  Makefile
hco_emislist_mod.F90     hcoio_write_std_mod.F90   messy_ncregrid_base.F90
hco_error_mod.F90        hco_logfile_mod.F90       messy_ncregrid_mpi.F90
hco_extlist_mod.F90      hco_readlist_mod.F90
hco_filedata_mod.F90     hco_restart_mod.F90
```

The **Core** directory contains code for:

- Defining HEMCO derived type objects
- Reading settings from configuration file
- Reading data from files
- Regridding horizontally (and vertically) to model grid
- Calculating emissions (applying masks/scale factors and combining inventories by hierarchy)

Questions?

Extra slides

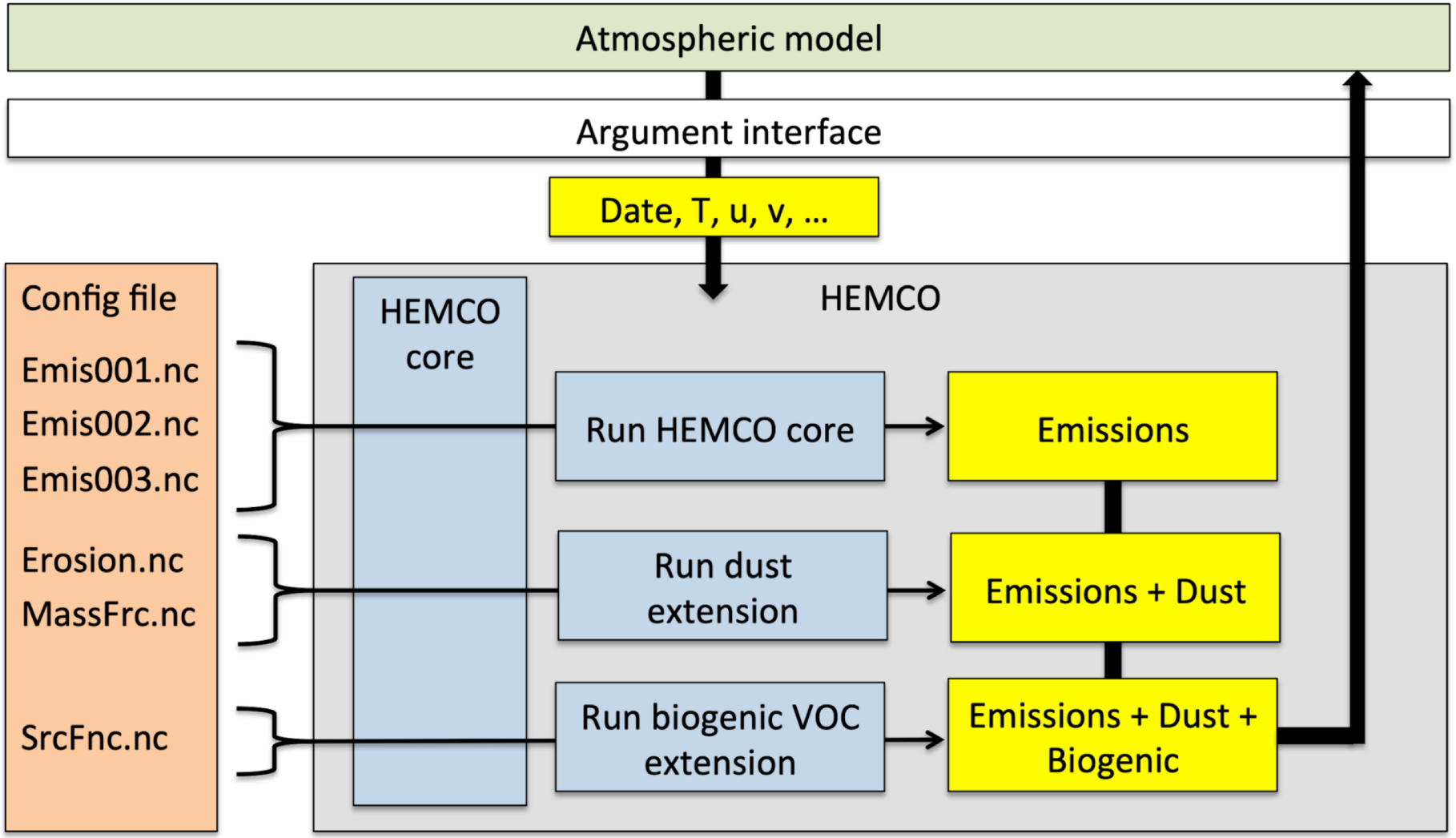
HEMCO/Extensions

```
Extensions:
CMakeLists.txt*              hcox_gc_POPs_mod.F90        hcox_soilnox_mod.F90
drydep_toolbox_mod.F90       hcox_gc_RnPbBe_mod.F90      hcox_state_mod.F90
hcox_ch4wetland_mod.F90      hcox_gfed_include_gfed3.H   hcox_template_mod.F90x
hcox_custom_mod.F90          hcox_gfed_include_gfed4.H   hcox_tomas_dustdead_mod.F
hcox_driver_mod.F90          hcox_gfed_mod.F90           hcox_tomas_jeagle_mod.F90
hcox_dustdead_mod.F          hcox_iodine_mod.F90         hcox_tools_mod.F90
hcox_dustginoux_mod.F90      hcox_lightnox_mod.F90       hcox_volcano_mod.F90
hcox_finn_include.FINNv16    hcox_megan_mod.F90          Makefile
hcox_finn_include.H          hcox_paranox_mod.F90        ocean_toolbox_mod.F90
hcox_finn_mod.F90            hcox_seaflux_mod.F90        Preprocess/
hcox_finn_mod.FINNv16        hcox_seasalt_mod.F90
```

The **Extensions** directory contains code for computing emissions that require knowledge about environmental fields. For example:

- Biogenic emissions: MEGAN
- Biomass burning emissions: GFED, FINN
- Dust emissions: DEAD or Ginoux schemes

HEMCO/Extensions



Keller et al. (2014)

HEMCO/Interfaces

```
Interfaces:
CMakeLists.txt*    hcoi_standalone_mod.F90  Makefile
hcoi_esmf_mod.F90  hemco_standalone.F90
```

The **Interfaces** directory contains code for running HEMCO in standalone mode or interfaced with other models

- The GEOS-Chem interface code is currently in the GEOS-Chem source code (in GeosCore/), but may be better suited here

Initialize

Interface (`HCOI_GC_Init`)

1. Print HEMCO version
2. Initialize **`HcoConfig`**
   - Sets met, grid, model species
3. Read configuration file
   - Phase 1: Read switches
   - Open log file
   - Phase 2: Read fields
4. Initialize **`HcoState`**
5. Call `HCO_Init`
6. Call `HCOX_Init`

HEMCO (`HCO_Init`)

1. Initializes `HcoState%AlltIDx`
2. Initializes `HcoState%Clock`
3. Initializes diagnostic collections (default, restart, manual)
4. Initialize `HcoState%ReadLists` and registers fields

Run

Interface (`HCOI_GC_Run`)

1. Phases:
   -1 : Used for GCHP (still true?)
   0 : Read initial met, BCs, and restart file
   1 : Update clock, data list, met fields, BCs
   2 : Calculate emissions
2. Call `HCO_Run`
3. GCC: Pass met, restart, BCs to model
4. Call `HCOX_Run`
5. Update diagnostics

HEMCO (`HCO_Run`)

1. Check if it's time for emissions
2. Write out diagnostics if it's time
3. Read/update data for Phase /=2 (`ReadList_Read`)
4. Calculate emissions if Phase = 2 (`Hco_CalcEmis`)

Finalize

Interface (`HCOI_GC_Final`)

1. Call `HCO_Final`
2. Call `HCOX_Final`
3. Clean up diagnostics
4. Clean up `HcoState`
5. Deallocate local arrays needed for interface

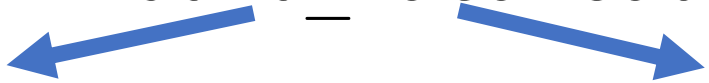
HEMCO (`HCO_Final`)

1. Write out diagnostics if needed
2. Cleanup derived type objects
3. Close HEMCO log file

File I/O

`ReadList_Read`

`HCOIO_DataRead`



MAPL_GetPointer
Point HEMCO to data (Lct%Dct%Dta%Arr2D%Val)

Populate FileData (Lct%Dct%Dta)
Open netCDF file
Select time from file *Get preferred time, apply time shift*, interpolate**
Read data from file
Convert units to kg/m2/s
Regridding (MESSy or A2A)
Update emission field (Lct%Dct%Dta%Arr2D%Val)

*If needed

Emissions Calculation

`HCO_CalcEmis`

Loop through all containers in `EmisList` *Sort by species, category, hierarchy*
Get current emissions and write to TmpFlx array *For each container apply scale factor(s) and mask(s)*
Apply species-specific scale factors
Combine emissions in same category based on hierarchy
Update emission field

Applying Emissions in GEOS-Chem Classic

`Do_Mixing`

Call `Do_Tend` to apply tendencies
Get emission fluxes and dry deposition frequencies from HEMCO via `GetHcoVal`
Apply fluxes to species concentration array `State_Chm%Species`

HEMCO derived types	Description
HcoState	Includes HcoGrid, HcoClock, HcoPhys, HcoMicroPhys, EmisList, ReadList, ESMF_GridComp, ESMF_State (Import+Export), etc.
HcoConfig	Configuration file settings
HcoSpc	HEMCO species information
VertGrid	Vertical grid definition (type, Ap, Bp)
HcoGrid	Horizontal grid definition used for interpolation (also includes PEDGE, PSfc BXHEIGHT, Zgrid)
HcoClock	HEMCO clock (current time, local time, previous time, simulation time)
HcoPhys	Physical constants (Avo, Pi, Re, AIRMW, g0, Rd, RSTARG)
HcoMicroPhys	Settings for aerosol microphysics (TOMAS)
ReadLists	Data containers used by HEMCO, organized by update frequency
DataCont	Stores information read from HEMCO config entries (name, Cat, Hier, ScalID, etc.)
ListCont	Used to create linked lists of DataCont
FileData	Holds information from source file data (filename, update frequency, resolution)
TimeIDx	Points to the current time slices of data within a file
EmisList	Containers needed for emission calculations sorted type, species, category, hierarchy
ExtState	Logical switches and settings for each extensions, includes pointers to met fields