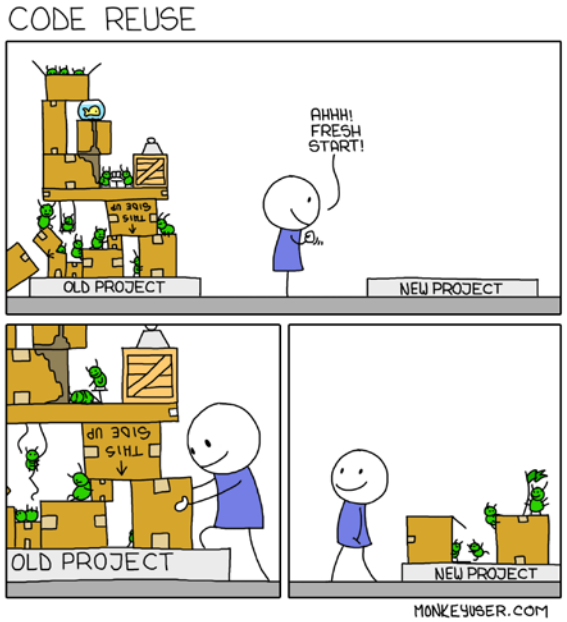
Harmonized Emissions Component (HEMCO)

Restructuring and interfacing with CAM

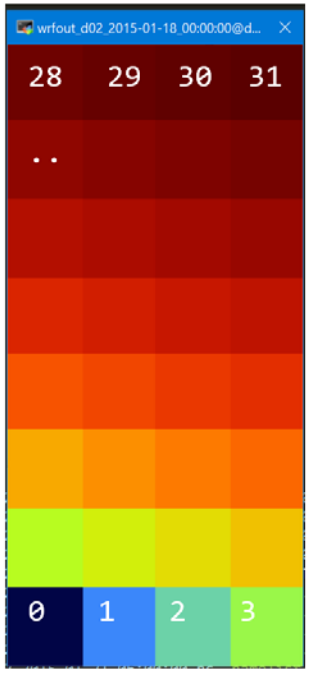
Haipeng Lin, Spring 2020

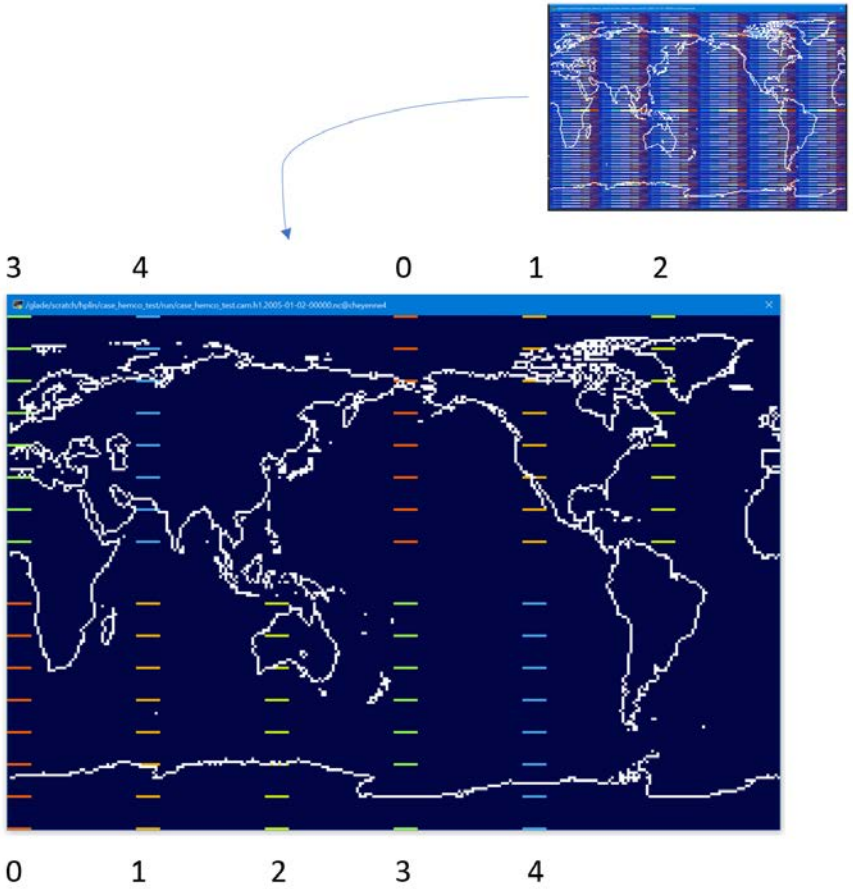


Tale of technical constraints

So I was explaining my proposed domain decomposition…

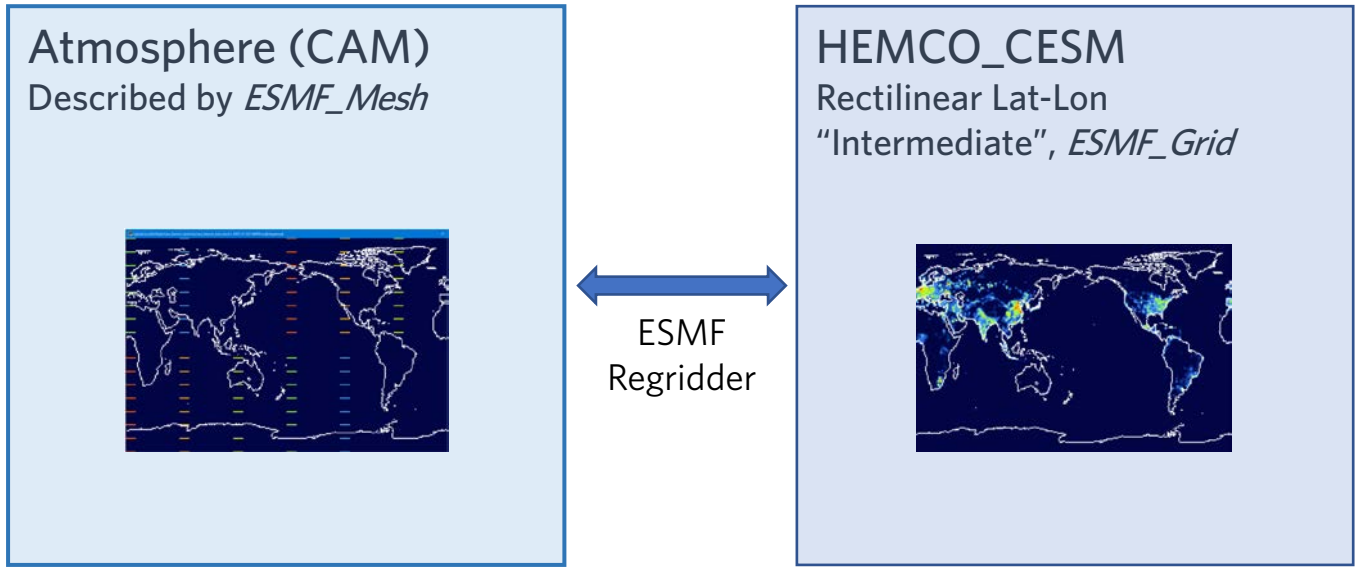
> “No, no, no. You live in that world. That is 10 years behind. The problem is, we’re in this world.”

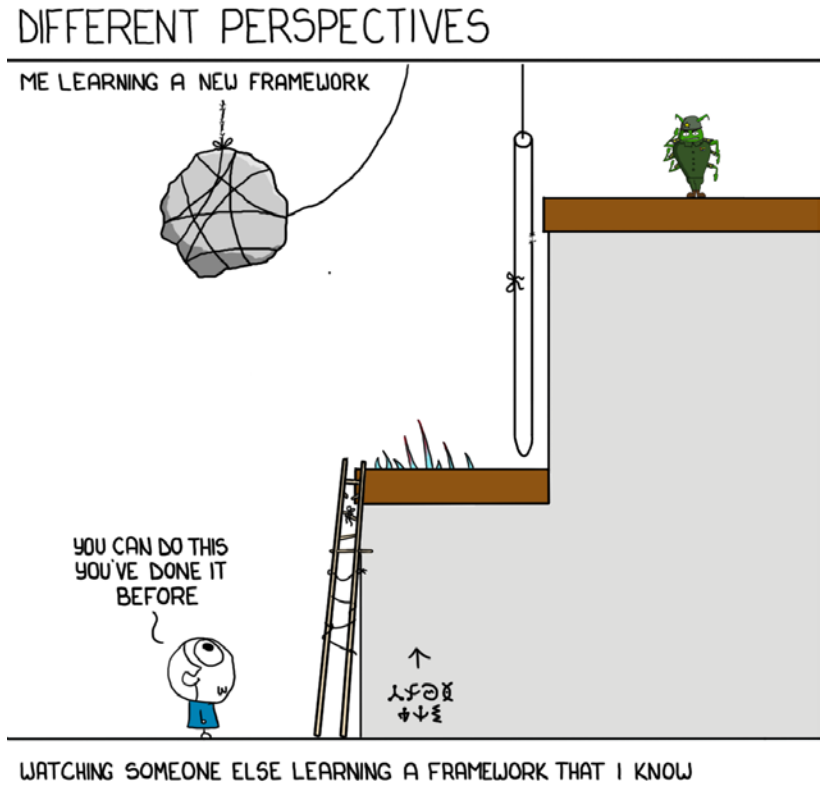




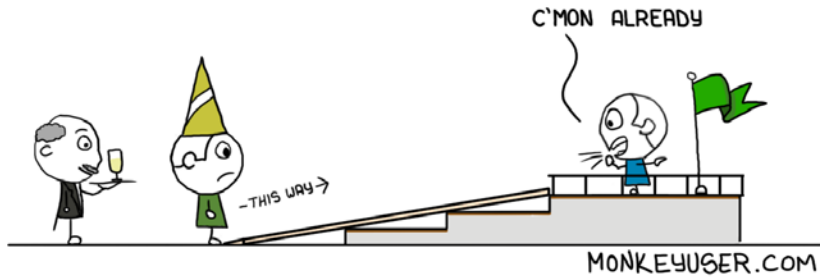
CAM | ESMF | HEMCO "Hybrid" Architecture

- Must introduce an ESMF-based regrid layer





- Regridder requires running in a Gridded Component
- But the model must not be driven by ESMF!



Building a new "HEMCO standalone"

Main components:

- HEMCO Driver

  "`hemco_interface.F90`"

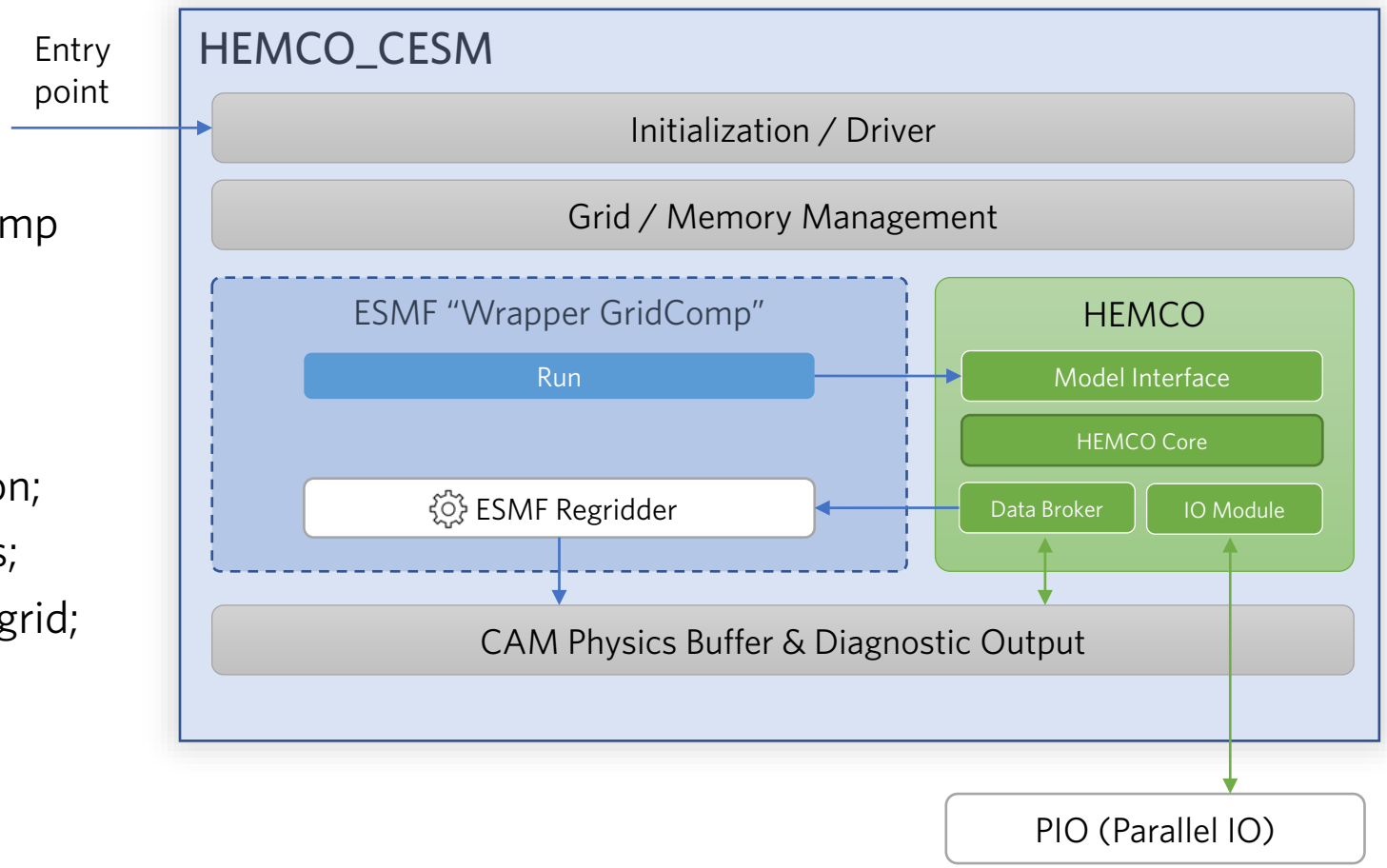
  - Creates HEMCO ESMF GridComp
  - "Controls ESMF"
- Grid Manager

  "`hco_esmf_grid.F90`"

  - Copies "`gc_grid_mod`" for lat-lon;
  - Manages ESMF Regrid Handles;
  - Performs HEMCO <-> CAM regrid;
- Export Manager

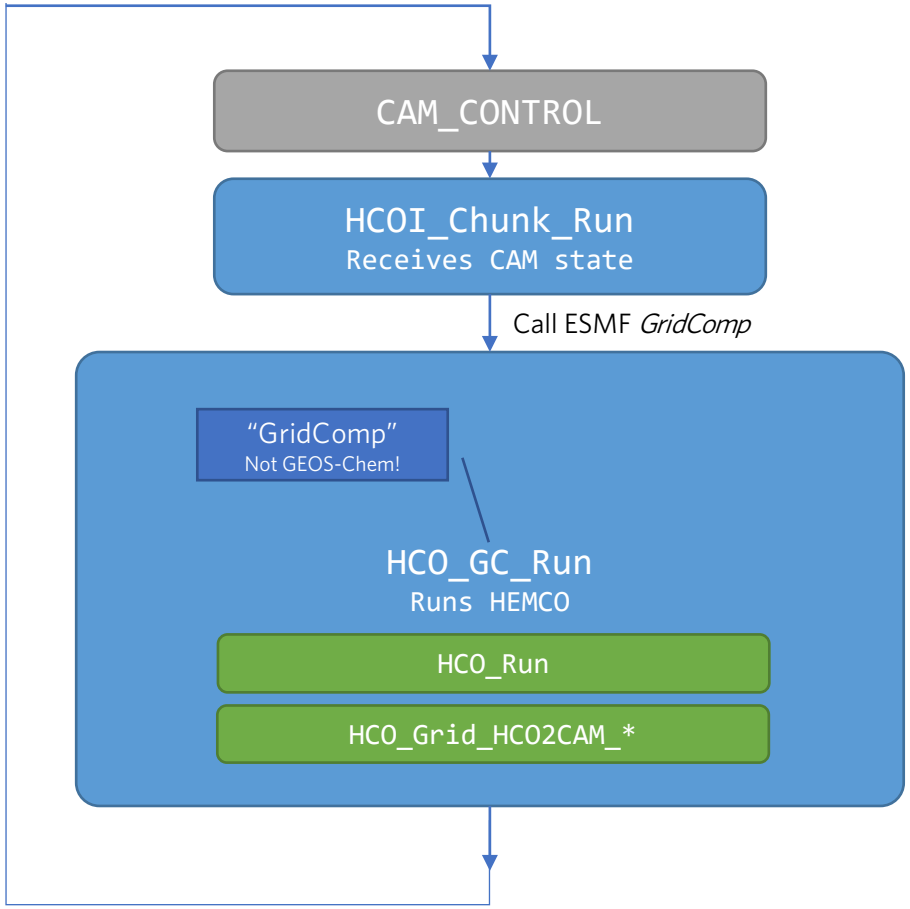
  "`hco_cam_exports.F90`"

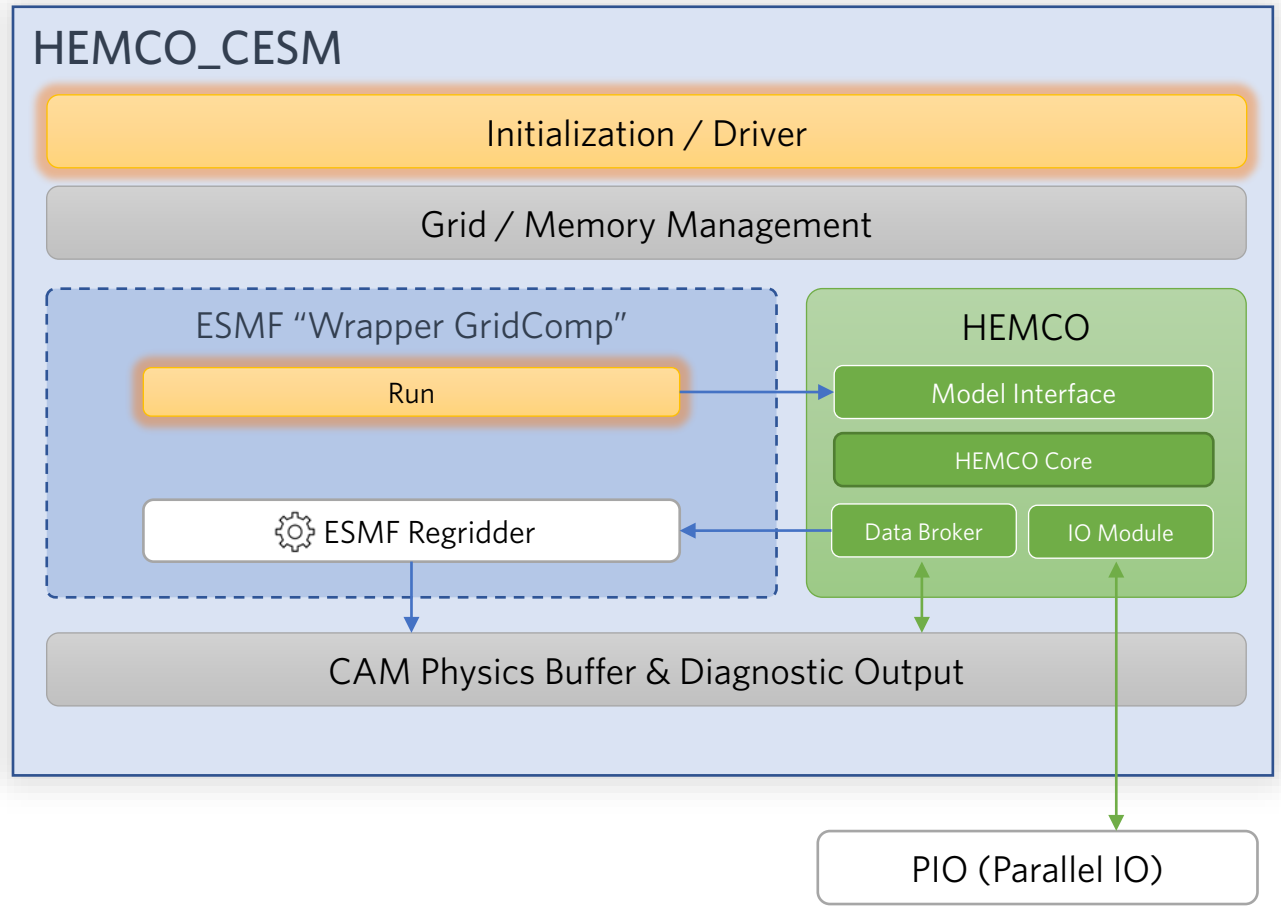
  - Exports to CAM History
  - Exports to CAM Physics Buffer



Building a new "HEMCO standalone"

What does building a model-in-a-model *really* look like?

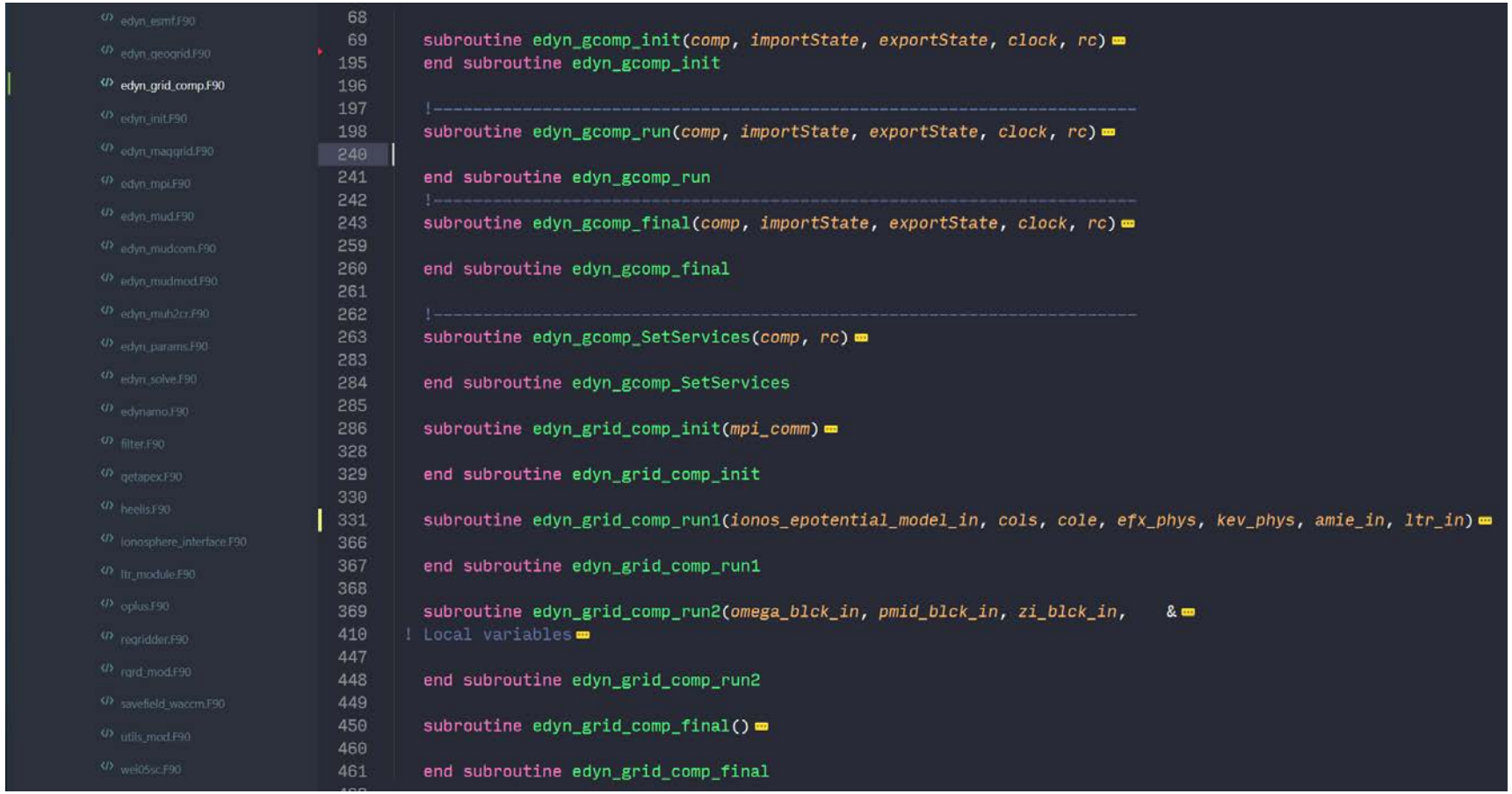




- HEMCO_CESM **operates on its own grid.**
  - Thus, HEMCO_CESM must handle its own domain decomposition within MPI.
- HEMCO_CESM is not *driven* by ESMF but ***drives* ESMF**!

I promise I did not invent this…

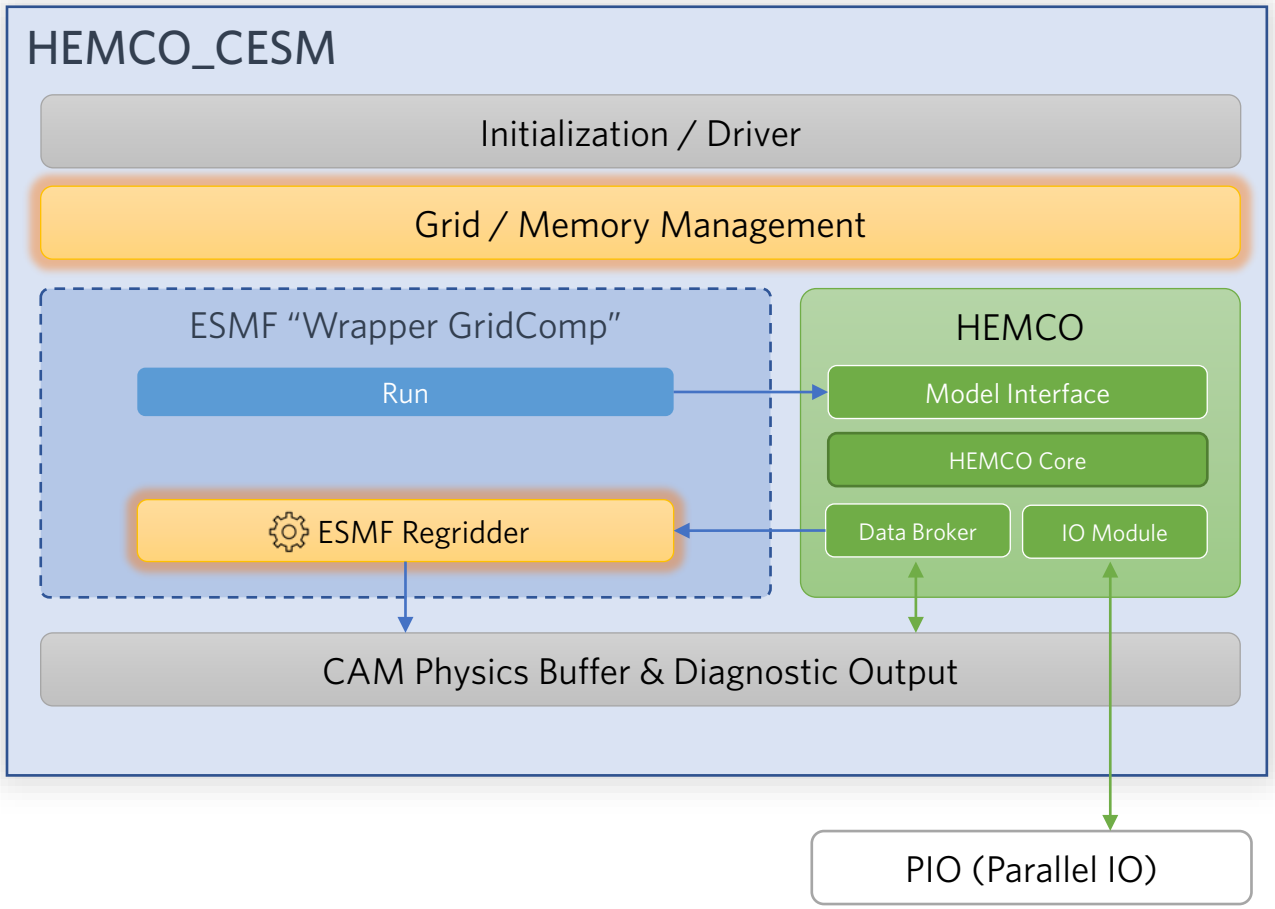
This implementation proudly brought to you by the CAM *WACCMX* interface



Building a new “HEMCO standalone”

The ESMF regridder

```
42  !
43  ! !PUBLIC MEMBER FUNCTIONS:
44  !
45      public        :: HCO_Grid_Init
46      public        :: HCO_Grid_UpdateRegrid
47
48      public        :: HCO_Grid_HCO2CAM_2D          ! Regrid HEMCO to CAM mesh on 2D field
49      public        :: HCO_Grid_HCO2CAM_3D          !                              ...on 3D field
50      public        :: HCO_Grid_CAM2HCO_2D          ! Regrid CAM mesh to HEMCO on 2D field
51      public        :: HCO_Grid_CAM2HCO_3D          !                              ...on 3D field
52
53  !
54  ! !PRIVATE MEMBER FUNCTIONS:
55  !
56      private       :: HCO_Grid_ESMF_CreateCAM
57      private       :: HCO_Grid_ESMF_CreateCAMField    ! Create field on CAM physics mesh
58      private       :: HCO_Grid_ESMF_CreateHCO         ! Create HEMCO lat-lon grid in ESMF
59      private       :: HCO_Grid_ESMF_CreateHCOField    ! Create field on HEMCO ll grid
60
61      private       :: HCO_ESMF_Set2DHCO               ! Set ESMF field with 2D HEMCO data
62      private       :: HCO_ESMF_Set3DHCO               ! Set ESMF field with 3D HEMCO data
63      private       :: HCO_ESMF_Set2DCAM               ! Set ESMF field with 2D CAM mesh data
64      private       :: HCO_ESMF_Set3DCAM               ! Set ESMF field with 3D CAM mesh data
65
66      private       :: HCO_ESMF_Get1DField             ! Retrieve 1D ESMF field pointer
67      private       :: HCO_ESMF_Get2DField             ! Retrieve 2D ESMF field pointer
68      private       :: HCO_ESMF_Get3DField             ! Retrieve 3D ESMF field pointer
```



Building a new “HEMCO standalone”

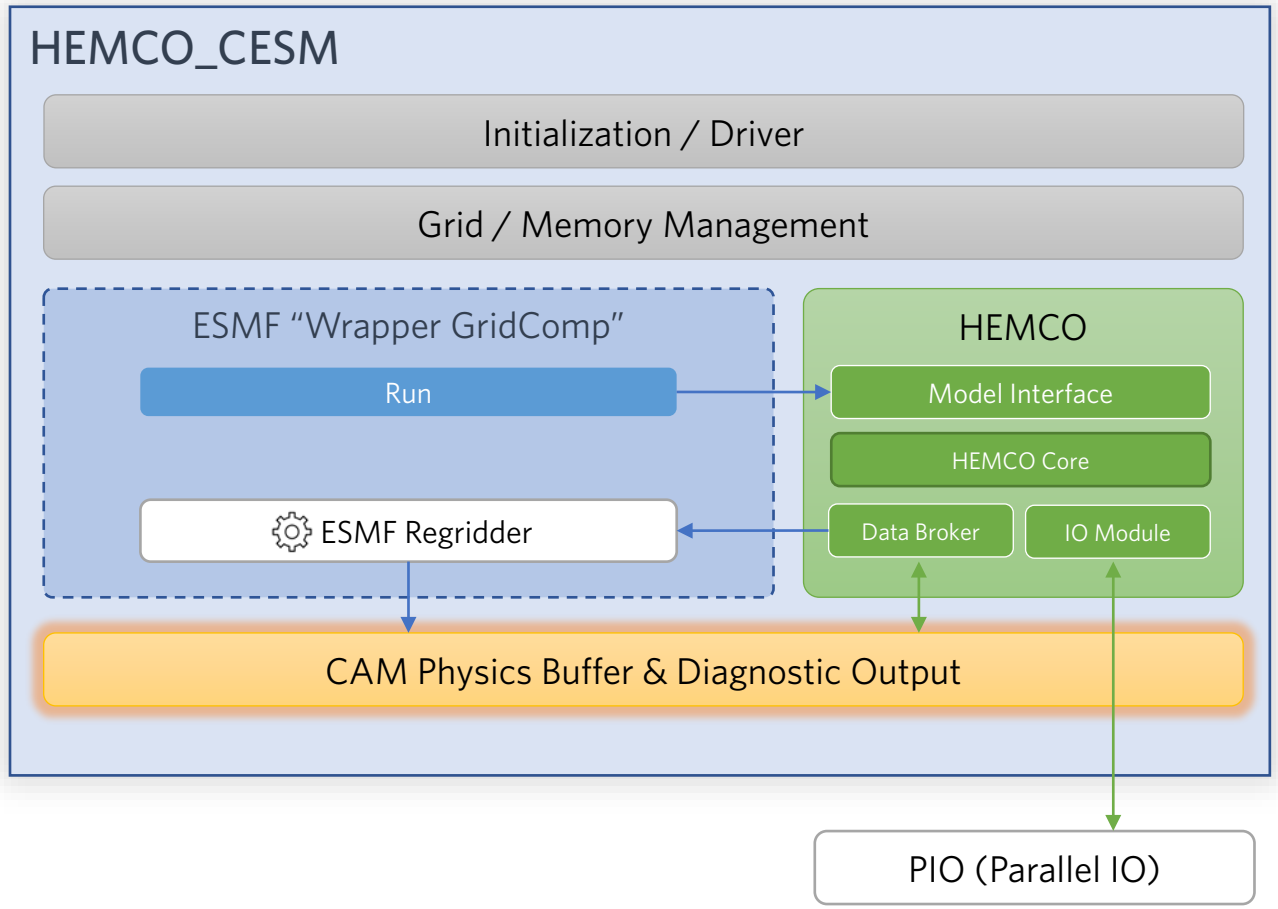
Interfacing with CAM from an exports component

```
48  !
49  ! !PUBLIC MEMBER FUNCTIONS:
50  !
51      public    :: HCO_Exports_Init
52      public    :: HCO_Export_History_HCO3D
53      public    :: HCO_Export_History_CAM3D
54
55      public    :: HCO_Export_Pbuf_AddField
56      public    :: HCO_Export_Pbuf_CAM3D
```

HEMCO_CESM is separate from all else!

- Runs in a “almost standalone” way (own grid, own memory)
- Writes emissions fluxes every timestep (30 minutes) to the “physics buffer”
- … to be retrieved in whatever way the chemistry sees fit.

Bonus: Easy compatibility



What does HEMCO need as input?

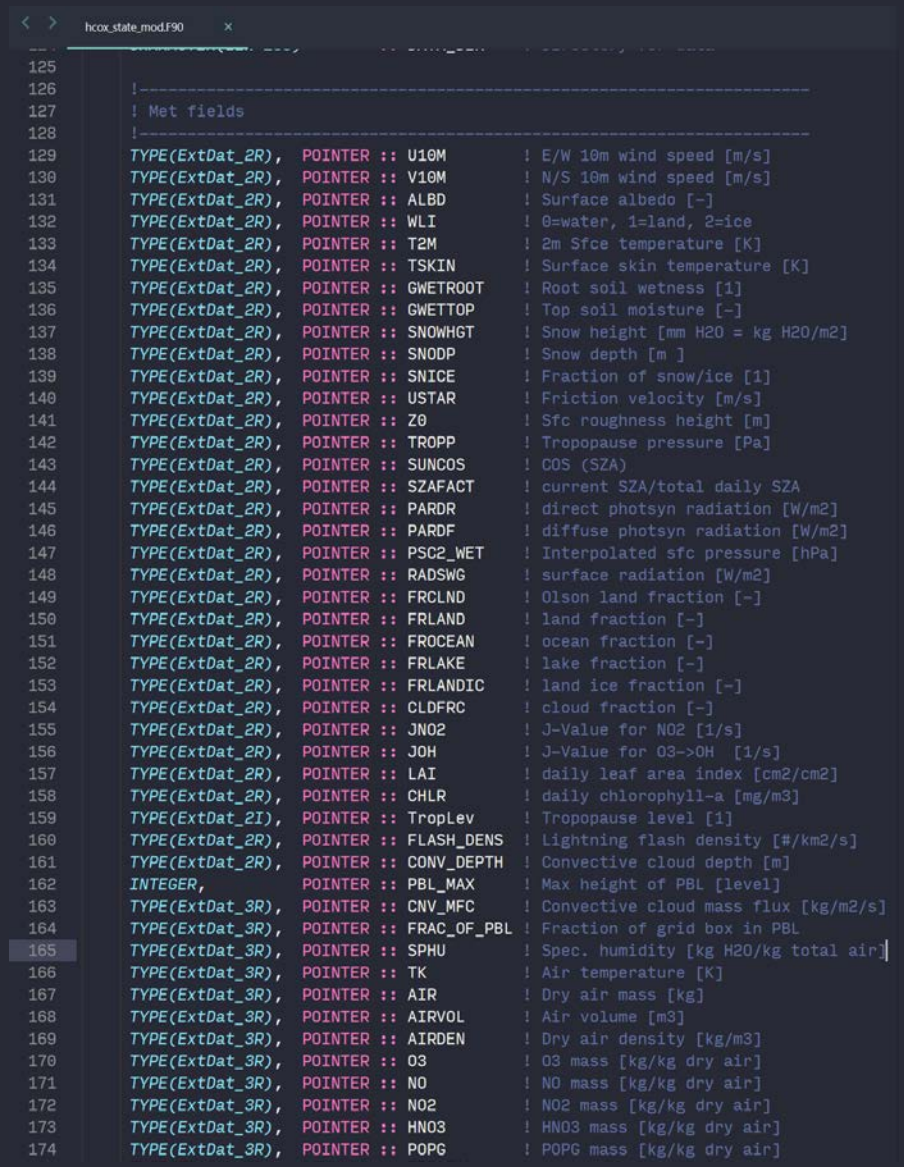
- Species Information
  - ID, name, molecular weight, "emMW_g", "MolecRatio", Henry's Law coef.
  - From comments:

```
! Emitted molecular weight of species [g/mol].
! Some hydrocarbon species (like ISOP) are emitted and
! transported as a number of equivalent carbon atoms.
! For these species, the emitted molecular weight will
! be 12.0 (the weight of 1 carbon atom).
HcoState%Spc(N)%EmMW_g     = SpcInfo%EmMw_g

! Emitted molecules per molecules of species [1].
! For most species, this will be 1.0.  For hydrocarbon
! species (like ISOP) that are emitted and transported
! as equivalent carbon atoms, this will be be the number
! of moles carbon per mole species.
HcoState%Spc(N)%MolecRatio = SpcInfo%MolecRatio
```

- Grid Information
- Meteorological input data
  - Necessitates "State Conversion" from (whatever model) → "GEOS-FP"
  - *Do we really need/want to use "GEOS-FP" for ExtState?*

Discussion: the “state conversion” and **code-reuse** issue



How to describe “Met Fields” for *HEMCO Extensions?*

- **Plan 1: “Keep as is”**
  - Use GEOS-FP met field format
  - Translate between “model native” & GEOS-FP using couplers
  - ✔ “Compatibility” with all extensions out-of-the-box
  - × How to handle code duplication?
- **Plan 2: Don’t impose extension met field format**
  - Need to create a more flexible “registry”
  - ✔ Avoids mess of “state translators” (or use “CCPP”)
  - × Cannot share extension sources between models

… other ideas welcome!

Discussion: where(when) should emissions be added?

From Thibaud:

Regarding the separation of HEMCO and mixing/drydep, I think that the most convenient way would probably be to add a field to State_Chm that would hold tendencies (State_Chm%Tendencies).

- *In GCC (and probably any other coupled model that uses HEMCO as a submodule?), this array would be filled through HEMCO as it is now. However it would be done outside of DO_TEND/DO_PBL_MIXING.*
- *In CESM-GC (where HEMCO lives at a higher level), the array could be populated in the interface (pp_geoschem/chemistry.F90) through calls to the physics buffer.*

Final Caveats

- Vertical regridding not working!
  - HCO_Interp_Mod only does 47 <-> 72 levels.
  - Should be able to do linear interpolation in pressure.
  - Will implement a lightweight "`REGRID_Z2Z_Mod`" to embed within HEMCO.
- Species list handling

The HEMCO-CESM project is made possible by:

- GCST and CESM® engineers and the GEOS-Chem ecosystem
- The CAM ionosphere interface
- The CESM2-GC project
- The ESMF framework