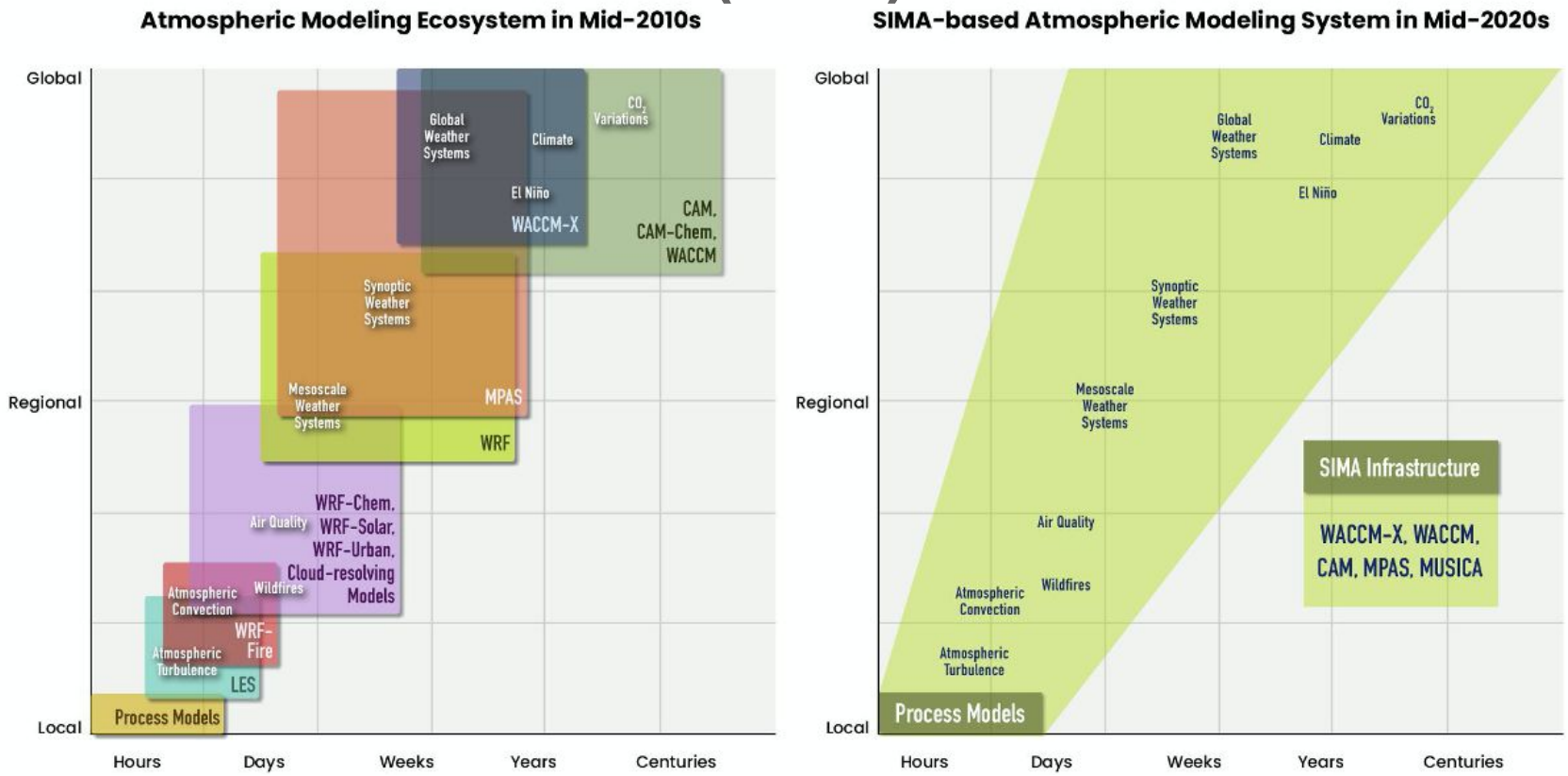
Implementing CCPP in CAM-SIMA

A tale of two (x2) repos

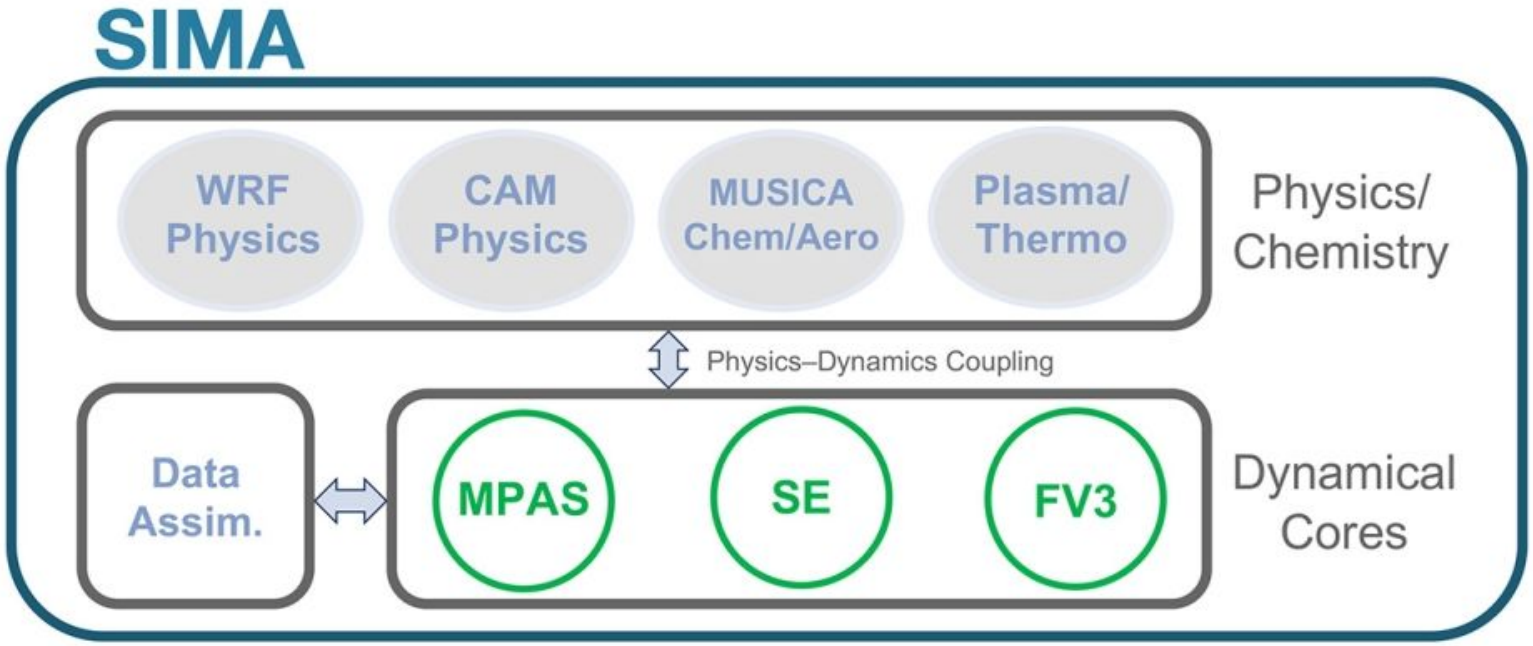
Jesse Nusbaumer,
Software Engineer, NCAR CGD-AMP

June 14th , 2023

System for Integrated Modeling of the Atmosphere (SIMA)

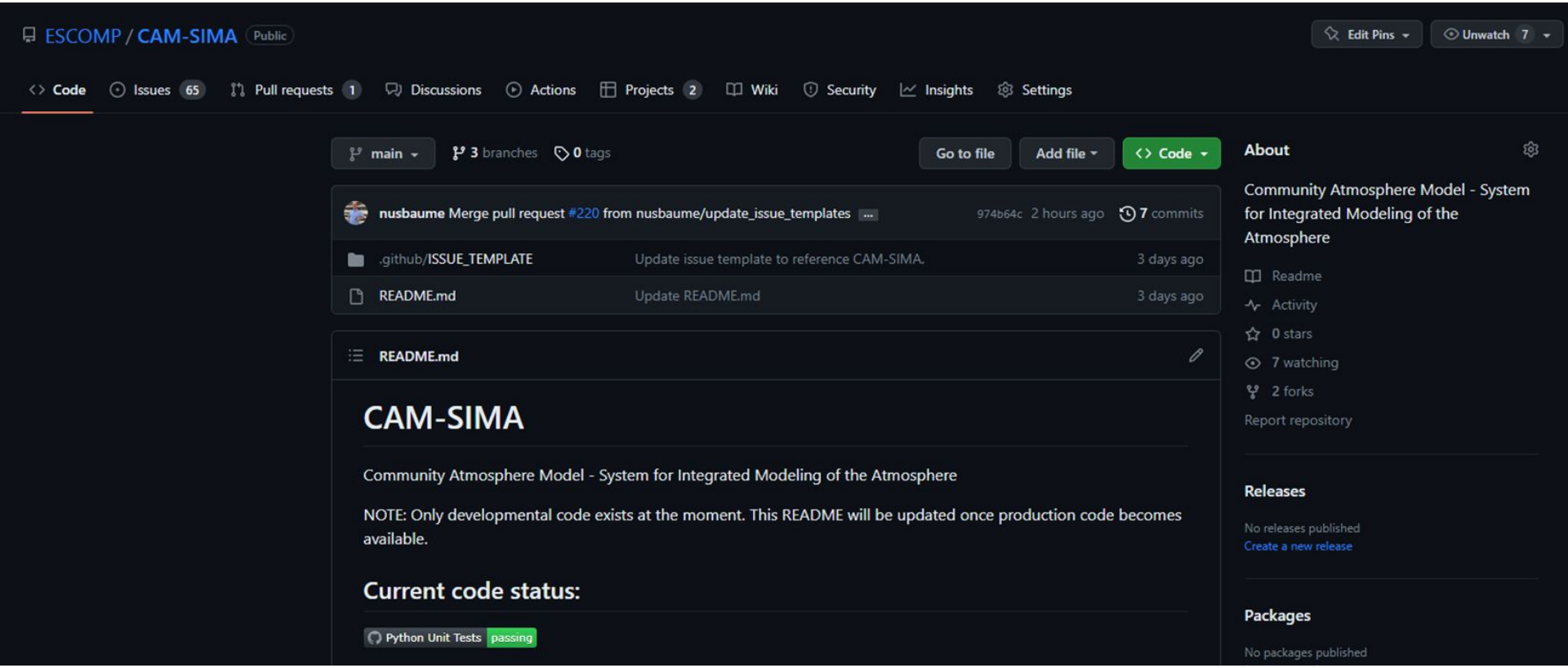


SIMA is a unified community atmospheric modeling framework, for use in an Earth System Model (ESM). SIMA enables diverse configurations of an atmosphere model inside of an ESM for applications spanning minutes to centuries and cloud to global scales, including atmospheric forecasts and projections of the atmospheric state and composition from the surface into the thermosphere.

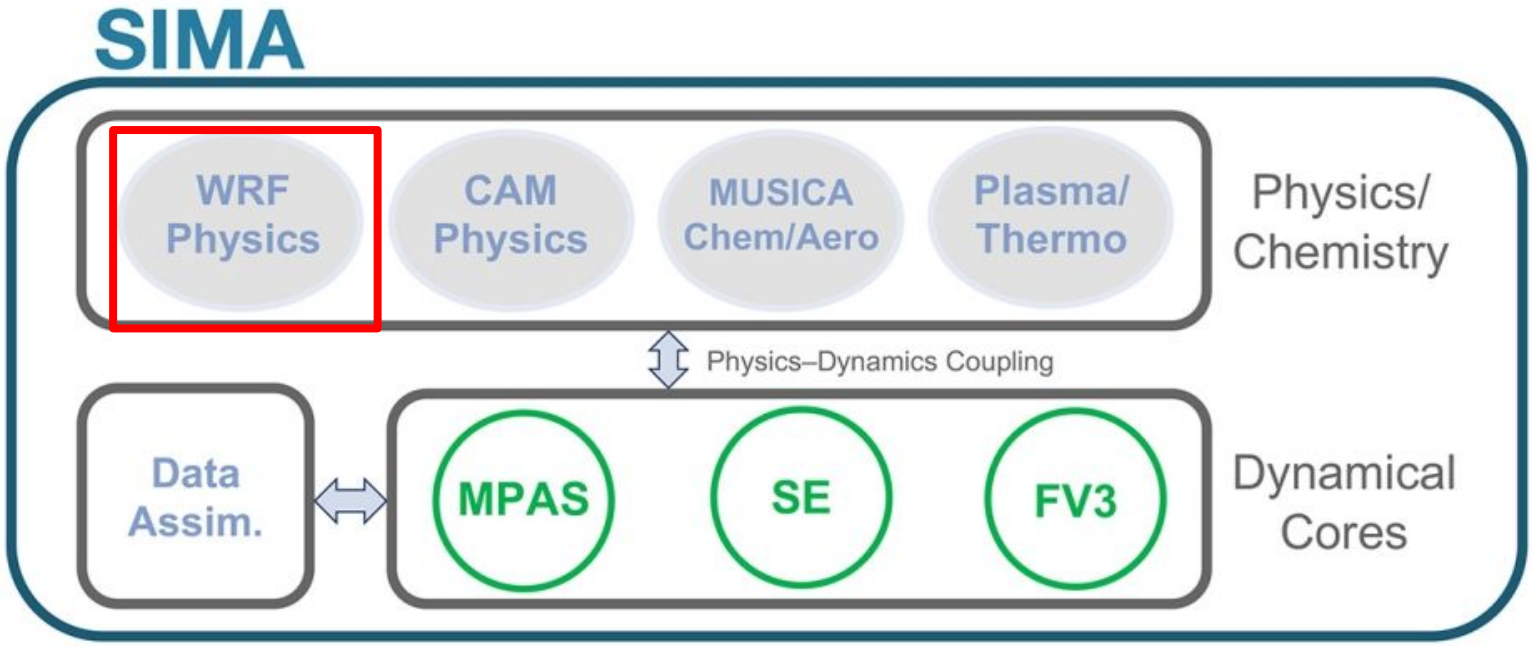


All of this is already in CAM in some form…

CAM-SIMA repo (repo #1)

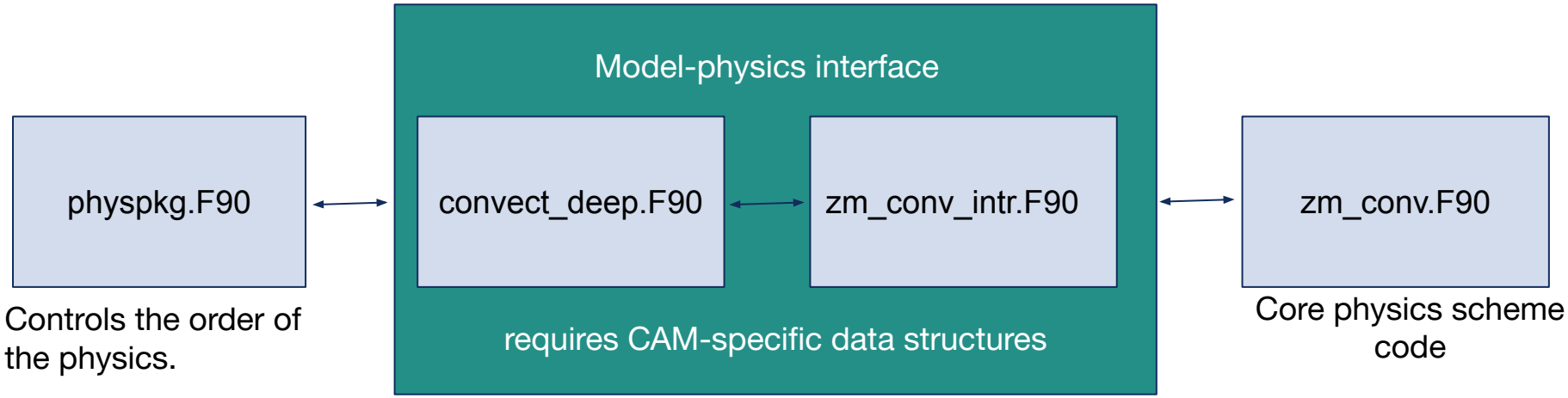


This code base used to be referred to as “CAMDEN”, or “new” CAM. Unlike the previous repo, this one is fully **public**.



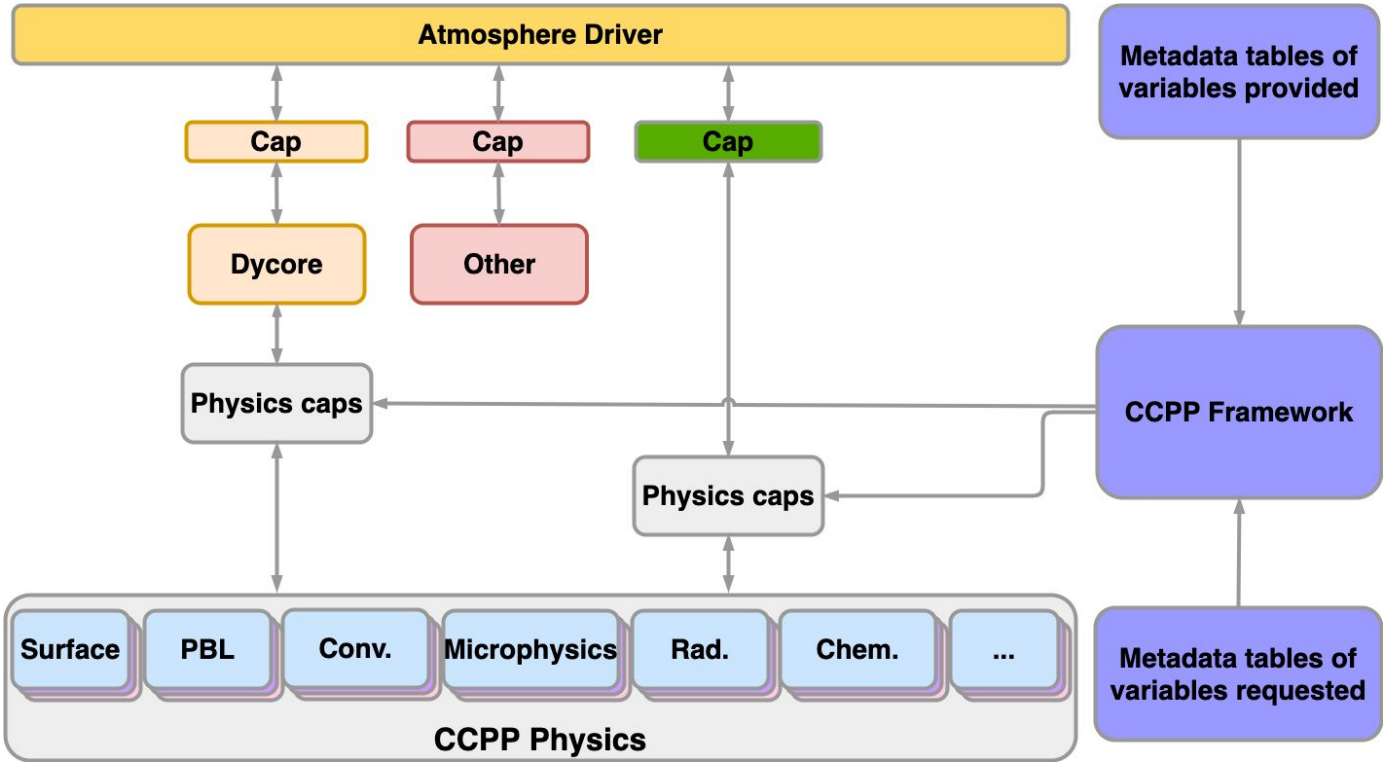
All of this is already in CAM in some form…*except* WRF Physics. How can we bring in entirely new sets of physics into what will eventually be SIMA?

Describe current CAM physics interface



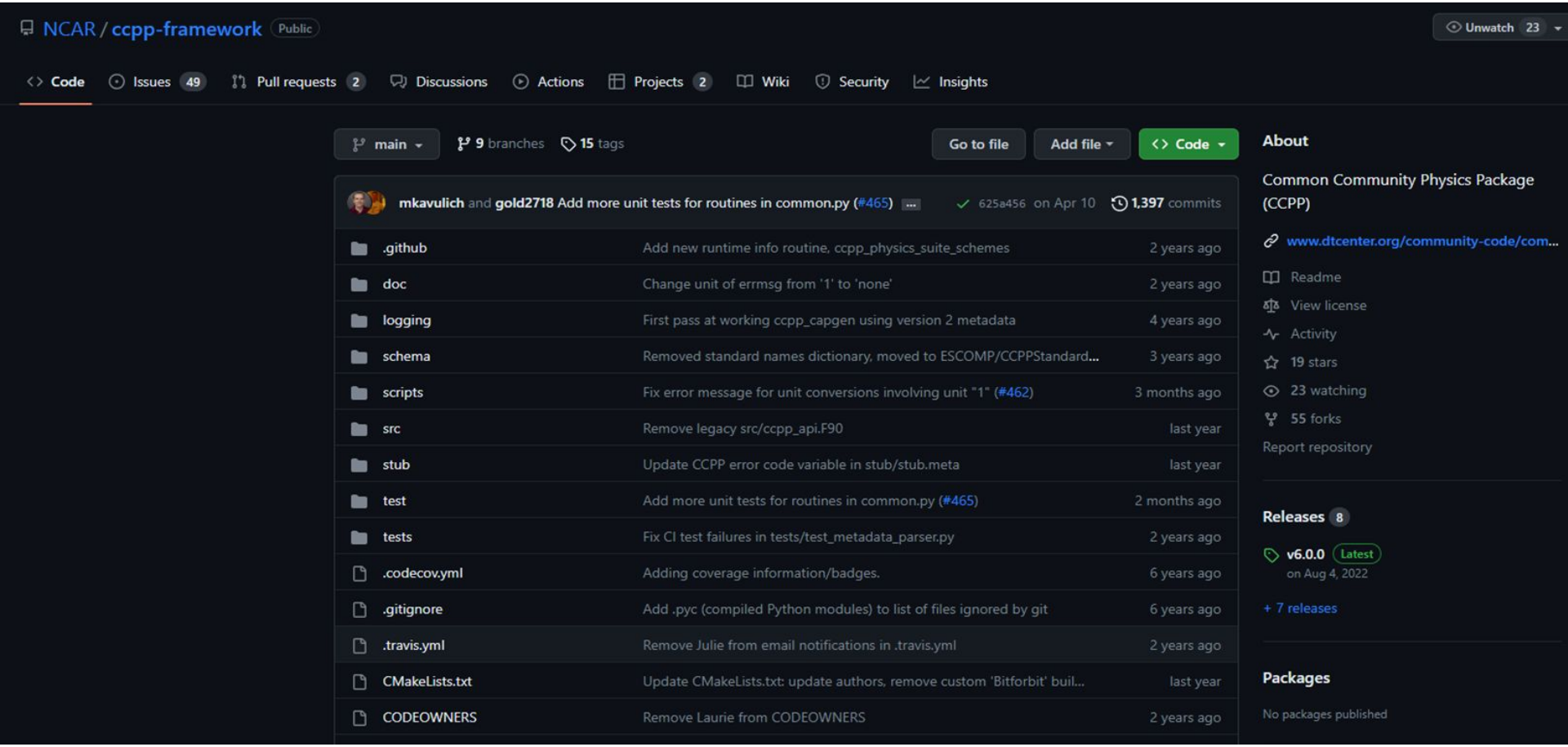
The files in the “Model-physics interface” would need to be created or modified, by hand, for every new WRF or NOAA physics scheme being brought into SIMA. Also different versions of “physpkg.F90” would need to be created

Common Community Physics Package (CCPP)



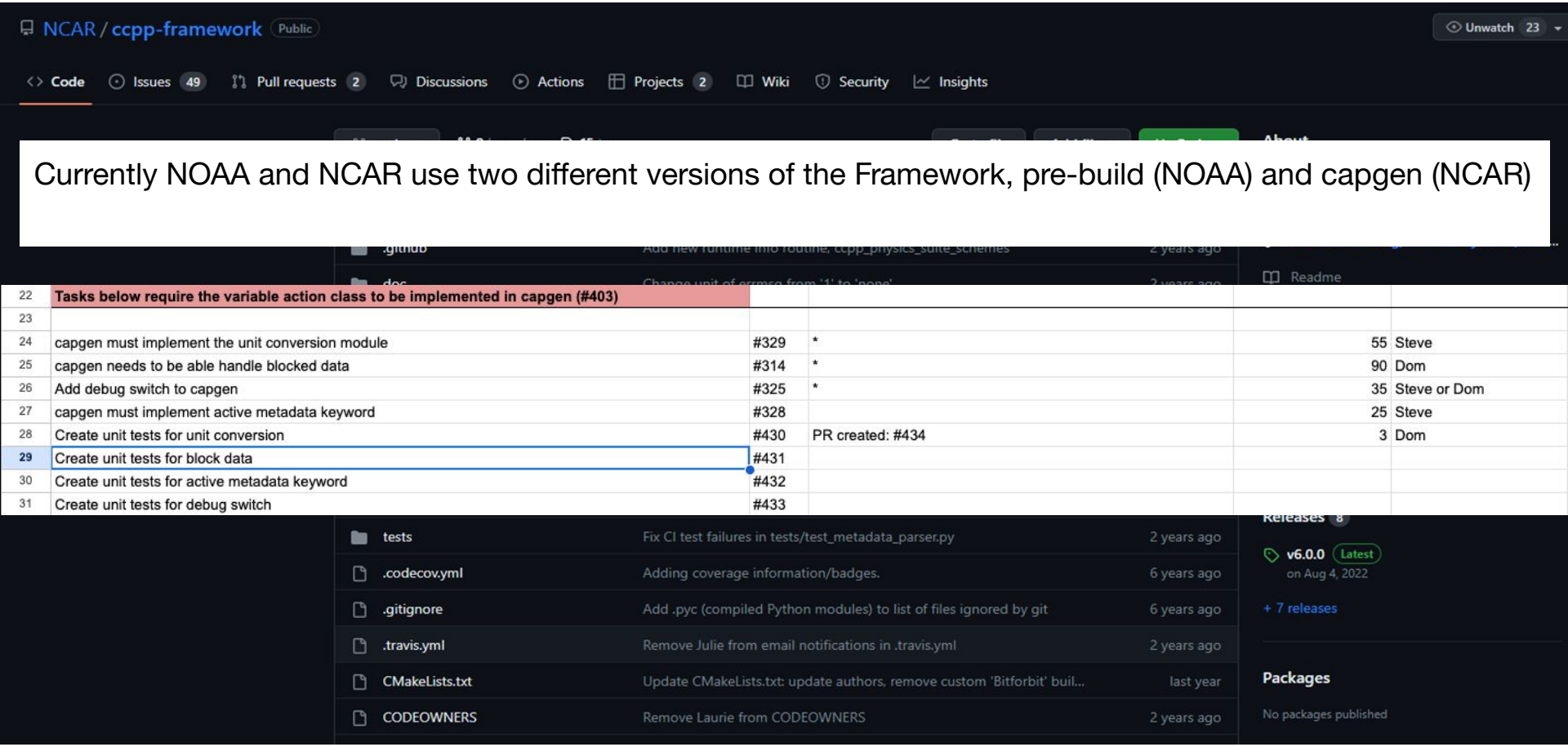
The CCPP is a software framework that automatically generates the Fortran interface (cap) layer for a physics parameterization (scheme).

The CCPP-Framework (Repo #2)



The CCPP-Framework (Repo #2)

Currently NOAA and NCAR use two different versions of the Framework, pre-build (NOAA) and capgen (NCAR)



22	**Tasks below require the variable action class to be implemented in capgen (#403)**				
23					
24	capgen must implement the unit conversion module	#329	*	55	Steve
25	capgen needs to be able handle blocked data	#314	*	90	Dom
26	Add debug switch to capgen	#325	*	35	Steve or Dom
27	capgen must implement active metadata keyword	#328		25	Steve
28	Create unit tests for unit conversion	#430	PR created: #434	3	Dom
29	Create unit tests for block data	#431			
30	Create unit tests for active metadata keyword	#432			
31	Create unit tests for debug switch	#433			

The CCPP-Framework (Repo #2)

Currently NOAA and NCAR use two different versions of the Framework, pre-build (NOAA) and capgen (NCAR)

NCAR / ccpp-framework Public

Unwatch 23

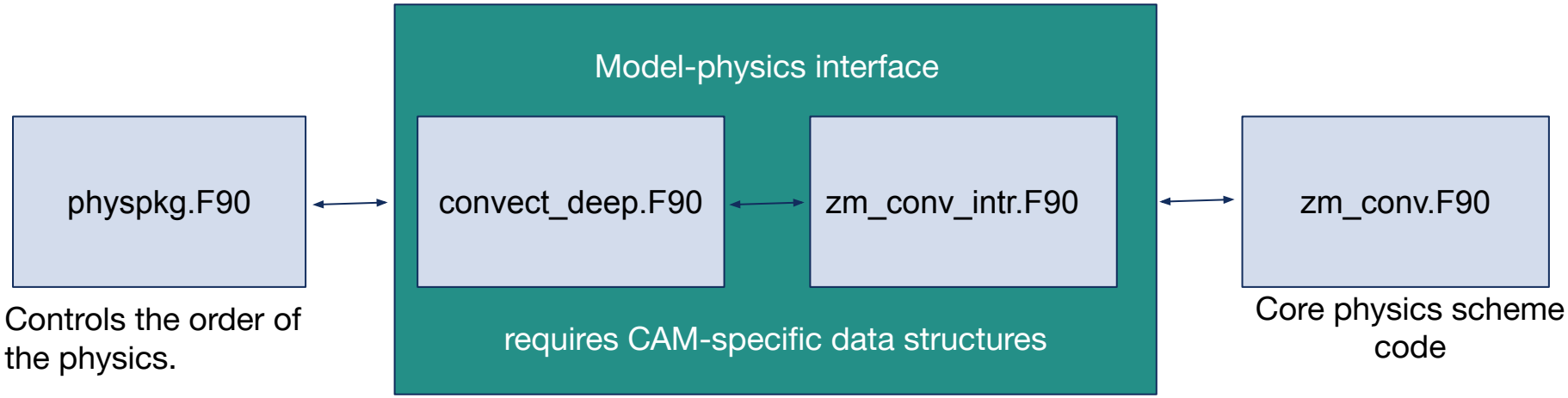
Code | Issues 49 | Pull requests 2 | Discussions | Actions | Projects 2 | Wiki | Security | Insights

22	**Tasks below require the variable action class to be implemented in capgen (#403)**				
23					
24	capgen must implement the unit conversion module	#329	*	55	Steve
25	capgen needs to be able handle blocked data	#314	*	90	Dom
26	Add debug switch to capgen	#325	*	35	Steve or Dom
27	capgen must implement active metadata keyword	#328		25	Steve
28	Create unit tests for unit conversion	#430	PR created: #434	3	Dom
29	Create unit tests for block data	#431			
30	Create unit tests for active metadata keyword	#432			
31	Create unit tests for debug switch	#433			

Dustin Swales

Courtney Peverley

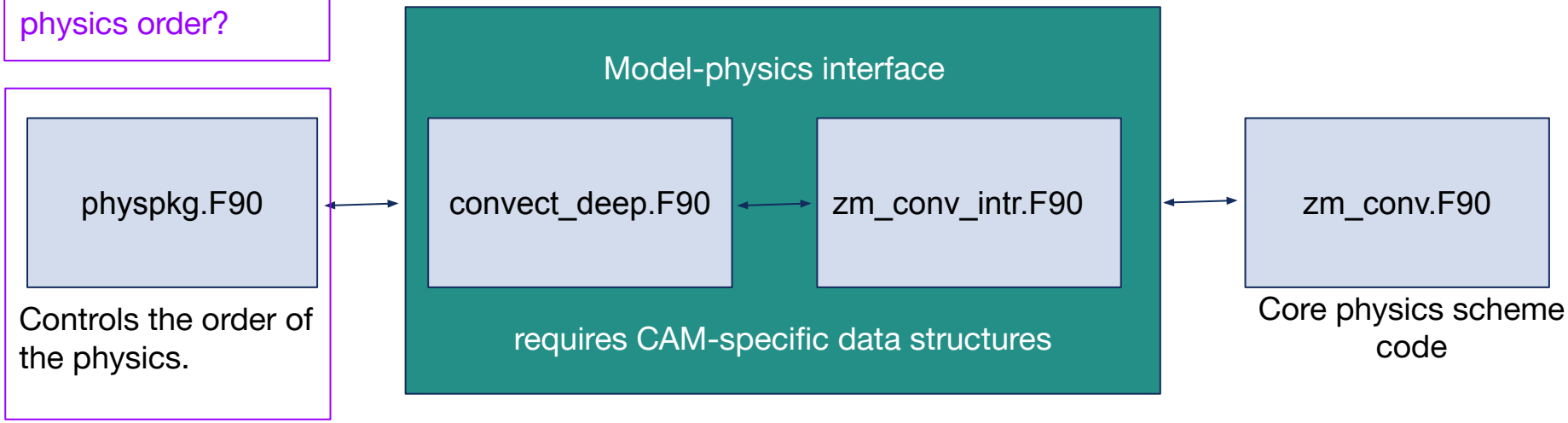
Describe current CAM physics interface



The files in the “Model-physics interface” would need to be created or modified, by hand, for every new WRF or NOAA physics scheme being brought into SIMA. Also different versions of “physpkg.F90” would need to be created

Describe current CAM physics interface

What's the CCPP's method to control physics order?



The files in the "Model-physics interface" would need to be created or modified, by hand, for every new WRF or NOAA physics scheme being brought into SIMA. Also different versions of "physpkg.F90" would need to be created

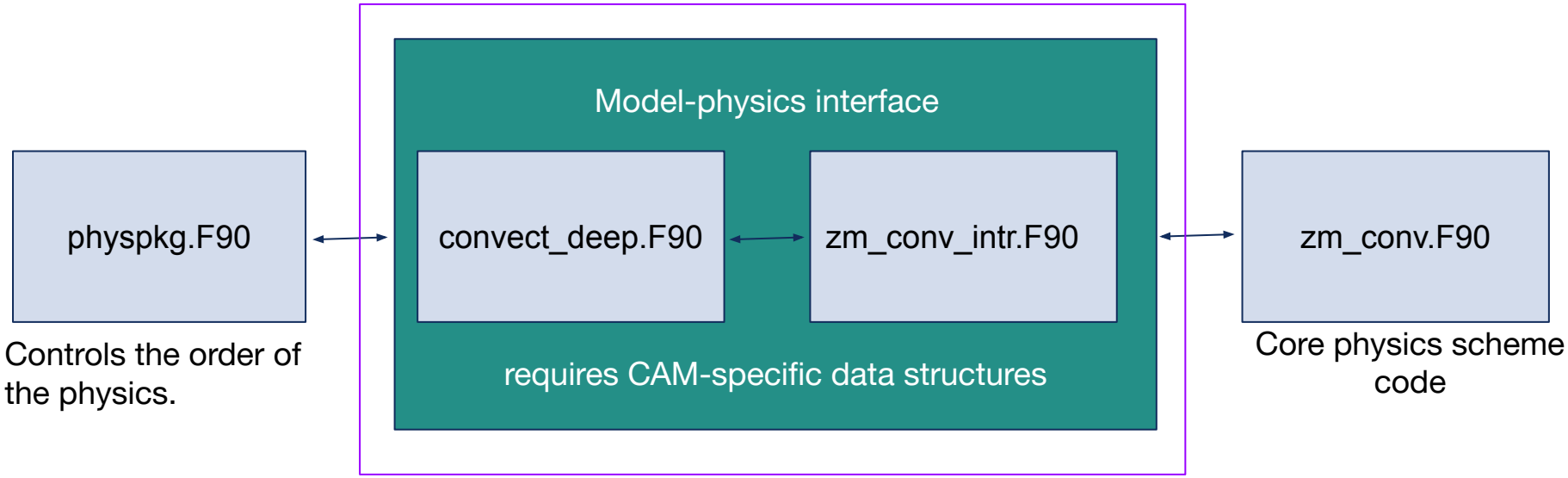
CCPP Suite Definition File

The list and order of physics schemes is controlled by a Suite Definition File (SDF), which allows for much easier re-ordering of physics routines, and removes the need to have a “physpkg.F90” source file.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <suite name="held_suarez_1994" version="1.0">
4    <group name="physics">
5      <scheme>held_suarez_1994</scheme>
6      <scheme>apply_tendency_of_x_wind</scheme>
7      <scheme>apply_tendency_of_y_wind</scheme>
8      <scheme>apply_heating_rate</scheme>
9      <scheme>qneg</scheme>
10   </group>
11 </suite>
```

Describe current CAM physics interface

What is needed for the CCPP to generate the interface?



The files in the “Model-physics interface” would need to be created or modified, by hand, for every new WRF or NOAA physics scheme being brought into SIMA. Also different versions of “physpkg.F90” would need to be created

CCPP Physics Scheme

Metadata (*.meta) file,
which lists metadata for all interface variables

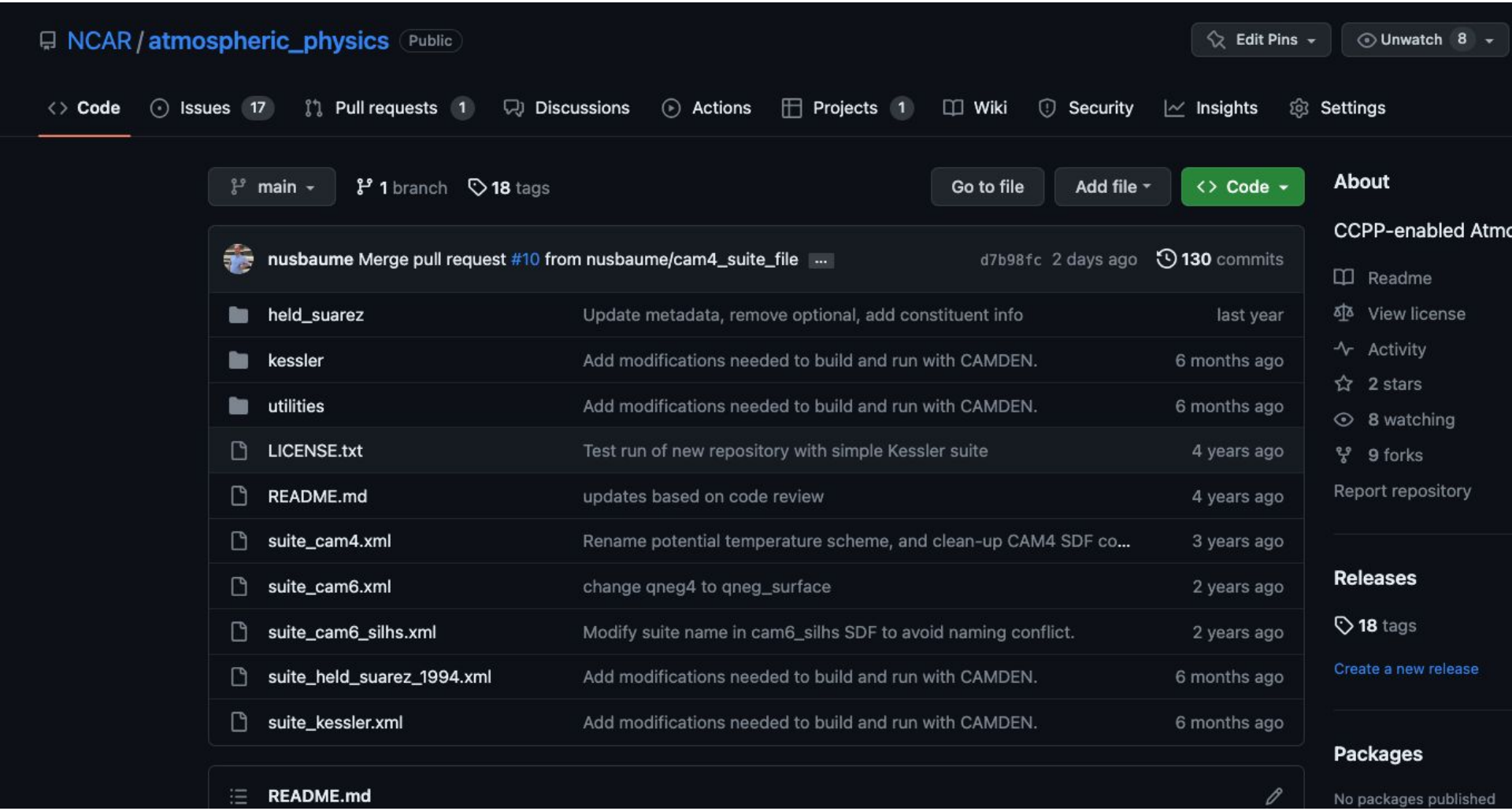
```
[ccpp-arg-table]
  name = apply_tendency_of_x_wind_run
  type = scheme
[ nz ]
  standard_name = vertical_layer_dimension
  long_name = Number of vertical layers
  units = count
  type = integer
  dimensions = ()
  intent = in
[ dudt ]
  standard_name = tendency_of_x_wind
  units = m s-2
  type = real | kind = kind_phys
  dimensions = (horizontal_loop_extent, vertical_layer_dimension)
  intent = in
[ u ]
  standard_name = x_wind
  units = m s-1
  type = real | kind = kind_phys
  dimensions = (horizontal_loop_extent, vertical_layer_dimension)
  intent = inout
  state_variable = True
[ dudt_total ]
  standard_name = tendency_of_x_wind_due_to_model_physics
  units = m s-2
  type = real | kind = kind_phys
  dimensions = (horizontal_loop_extent, vertical_layer_dimension)
  intent = inout
```

Source code (*.F90) file,
which contains the actual parameterization code

```
!> \section arg_table_apply_tendency_of_x_wind_run  Argument Table
! \htmlinclude apply_tendency_of_x_wind_run.html
subroutine apply_tendency_of_x_wind_run(nz, dudt, u, dudt_total, dt, &
                                        errcode, errmsg)

   ! Dummy arguments
   integer,                intent(in)    :: nz               ! Num vertical  layers
   real(kind_phys),    intent(in)    :: dudt(:,:)        ! tendency of x wind
   real(kind_phys),    intent(inout) :: u(:,:)           ! x wind
   real(kind_phys),    intent(inout) :: dudt_total(:,:) ! total tendency of x wind
   real(kind_phys),    intent(in)    :: dt               ! physics time step
   integer,            intent(out)   :: errcode
   character(len=512), intent(out)   :: errmsg

   ! Local variable
   integer :: klev

   errcode = 0
   errmsg = ''

   do klev = 1, nz
      u(:, klev) = u(:, klev) + (dudt(:, klev) * dt)
      dudt_total(:, klev) = dudt_total(:, klev) + dudt(:, klev)
   end do

end subroutine apply_tendency_of_x_wind_run
```

With these two files and a host model metadata file,
the model/scheme interface can be auto-generated

atmospheric_physics (repo #3)



CCPP Standard Names

```
[ccpp-arg-table]
  name = apply_tendency_of_x_wind_run
  type = scheme
[ nz ]
  standard_name = vertical_layer_dimension
  long_name = Number of vertical layers
  units = count
  type = integer
  dimensions = ()
  intent = in
[ dudt ]
  standard_name = tendency_of_x_wind
  units = m s-2
  type = real | kind = kind_phys
  dimensions = (horizontal_loop_extent, vertical_layer_dimension)
  intent = in
[ u ]
  standard_name = x_wind
  units = m s-1
  type = real | kind = kind_phys
  dimensions = (horizontal_loop_extent, vertical_layer_dimension)
  intent = inout
  state_variable = True
[ dudt_total ]
  standard_name = tendency_of_x_wind_due_to_model_physics
  units = m s-2
  type = real | kind = kind_phys
  dimensions = (horizontal_loop_extent, vertical_layer_dimension)
  intent = inout
```

CCPP “standard names” are the official name of a particular variable, and are what the CCPP-framework uses to determine what host model or physics scheme variable should be passed to another physics scheme.

Standard Name example discussion

- Q

Standard Name example discussion

- Q
- Q -> specific_humidity

Standard Name example discussion

- Q
- Q -> specific_humidity
- Q -> water_vapor_mixing_ratio_wrt_moist_air

Standard Name example discussion

- Q
- Q -> specific_humidity
- Q -> water_vapor_mixing_ratio_wrt_moist_air
- Q -> water_vapor_mixing_ratio_wrt_total_mass

Standard Name example discussion

- Q
- Q -> specific_humidity
- Q -> water_vapor_mixing_ratio_wrt_moist_air
- Q -> water_vapor_mixing_ratio_wrt_total_mass
- Q -> water_vapor_mixing_ratio_wrt_moist_air_and_condensed_water

CCPP Standard Names

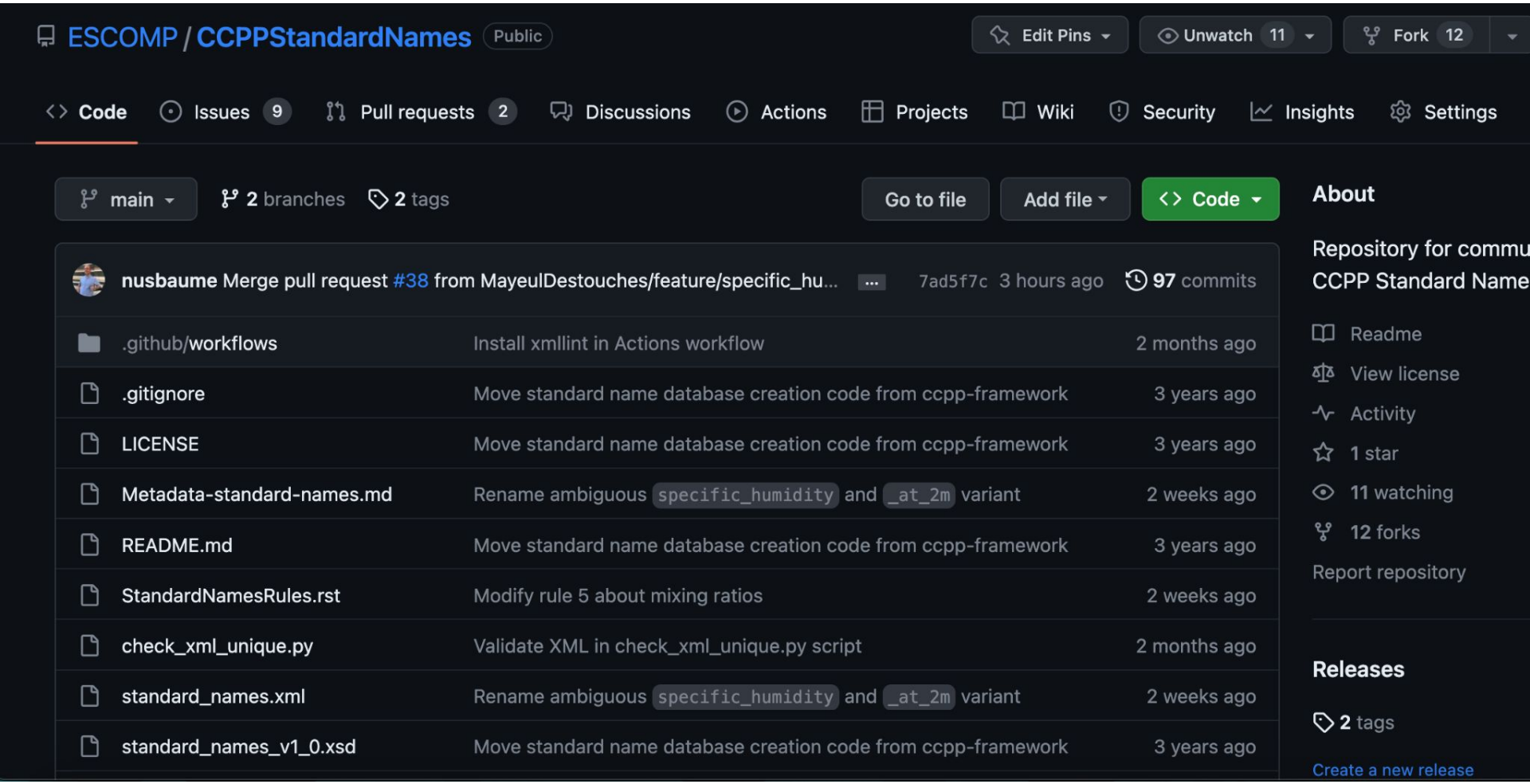
1	Snapshot Variable Name	Snapshot Longname	Longname Location	Snapshot Units	CCPP Standard Name	Accepted
400	ptend_taux_top	net zonal stress at top of model	physics_types.F90: line 163	Pa	N/A	
401	ptend_tauy_srf	net meridional stress at surface	physics_types.F90: line 164	Pa	N/A	
402	ptend_tauy_top	net meridional stress at top of model	physics_types.F90: line 165	Pa	N/A	
403					**TPHYSAC VARIABLES**	
404	tphysac_cmfmc	convective mass flux (m sub c)	physpkg.F90: line 2072		atmosphere_convective_mass_flux_due_to_all_convection	X
405	tphysac_det_ice	vertical integral of detrained ice	physpkg.F90: line 2141		vertically_integrated_detrainment_of_ice_due_to_all_convection	X
406	tphysac_det_s	vertical integral of detrained static energy from ice	physpkg.F90: line 2140		vertically_integrated_heating_from_freezing_of_detrained_liquid_due_to_all_convection	X
407	tphysac_dlf	detraining cld H2O from shallow + deep convections	physpkg.F90: line 2076		detrainment_of_water_due_to_all_convection	X
408	tphysac_dlf2	detraining cld H2O from shallow convections	physpkg.F90: line 2077		detrainment_of_water_due_to_shallow_convection	X
409	tphysac_fh2o	h2o flux to balance source from methane chemistry	physpkg.F90: line 1433		vertically_integrated_water_flux_due_to_chemistry	X
410	tphysac_fix_heat	heat flux for check_energy_chng	physpkg.F90: line 1434			
411	tphysac_net_flx		physpkg.F90: line 2069		net_radiative_fluxes_through_top_and_bottom_of_atmosphere_column	X
412	tphsyac_obklen	Obukhov length	physpkg.F90: line 1432		obukhov_length	X
413	tphysac_rliq	vertical integral of liquid not yet in q (ixcldliq)	physpkg.F90: line 2137	kg m-2 s-1	vertically_integrated_cloud_liquid_tendency_due_to_all_convection_to_be_applied_later_in_time_loop	X
414	tphysac_rliq2	vertical integral of liquid from shallow scheme	physpkg.F90: line 2139		vertically_integrated_cloud_liquid_tendency_due_to_shallow_convection_to_be_applied_later_in_time_loop	X
415	tphysac_surfric	surface friction velocity	physpkg.F90: line 1431		surface_layer_friction_velocity	X
416	tphysac_zdu	detraining mass flux from deep convection	physpkg.F90: line 2071		detrainment_mass_flux_due_to_deep_convection	X
417					**TPHYSBC VARIABLES**	
418	tphysbc_cmfcme	cmf condensation - evaporation	physpkg.F90: line 2074		condensation_minus_evaporation_due_to_deep_convection	X
419	tphysbc_cmfmc	convective mass flux (m sub c)	physpkg.F90: line 2072		atmosphere_convective_mass_flux_due_to_all_convection	X
420	tphysbc_dlf	detraining cld H2O from shallow + deep convections	physpkg.F90: line 2076		detrainment_of_water_due_to_all_convection	X
421	tphysbc_dlf2	detraining cld H2O from shallow convections	physpkg.F90: line 2077		detrainment_of_water_due_to_shallow_convection	X
422	tphysbc_fix_heat	heat flux for check_energy_chng	physpkg.F90: line 1434		surface_upward_heat_flux_in_air, surfa	
423	tphysbc_net_flx		physpkg.F90: line 2069		net_radiative_fluxes_through_top_and_bottom_of_atmosphere_column	X
424	tphysbc_pflx	conv rain flux throughout bottom of lev	physpkg.F90: line 2078		precipitation_flux_at_interface_due_to_deep_convection	X
425	tphysbc_rice	vertical integral of ice not yet in q (ixcldice)	physpkg.F90: line 2138		vertically_integrated_cloud_ice_tendency_due_to_all_convection_to_be_applied_later_in_time_loop	X
426	tphysbc_rliq	vertical integral of liquid not yet in q (ixcldliq)	physpkg.F90: line 2137		vertically_integrated_cloud_liquid_tendency_due_to_all_convection_to_be_applied_later_in_time_loop	X
427	tphysbc_rliq2	vertical integral of liquid from shallow scheme	physpkg.F90: line 2139		vertically_integrated_cloud_liquid_tendency_due_to_shallow_convection_to_be_applied_later_in_time_loop	X
428	tphysbc_zdu	detraining mass flux from deep convection	physpkg.F90: line 2071		detrainment_mass_flux_due_to_deep_convection	X
429						
430					**NCAR/ATMOSPHERIC_PHYSICS**	
431	**Variable Name**	**CAM equivalent variable**	**Meta File**		**CCPP Standard Name**	
432	rho	No CAM equivalent	kessler.meta		density_of_dry_air	X
433	qr	state%q(:,:,ixrain)	kessler.meta		rain_mixing_ratio_wrt_dry_air	X
434	scheme_name	No CAM equivalent	kessler.meta		scheme_name	X
435	rair	rairv	geopotential_t.meta		composition_dependent_gas_constant_of_dry_air	X
436	zvir	zvirv	geopotential_t.meta		ratio_of_water_vapor_gas_constant_to_composition_dependent_dry_air_gas_constant_minus_one	X
437	zi	state%zi	geopotential_t.meta		geopotential_height_wrt_surface_at_interface	X
438	dudt	ptend%u	physics_tendency_updaters.meta		tendency_of_eastward_wind	
439	dudt_total	tend%u	physics_tendency_updaters.meta		tendency_of_eastward_wind_due_to_model_physics	
440	dvdt	ptend%v	physics_tendency_updaters.meta		tendency_of_northward_wind	

CCPP Standard Names

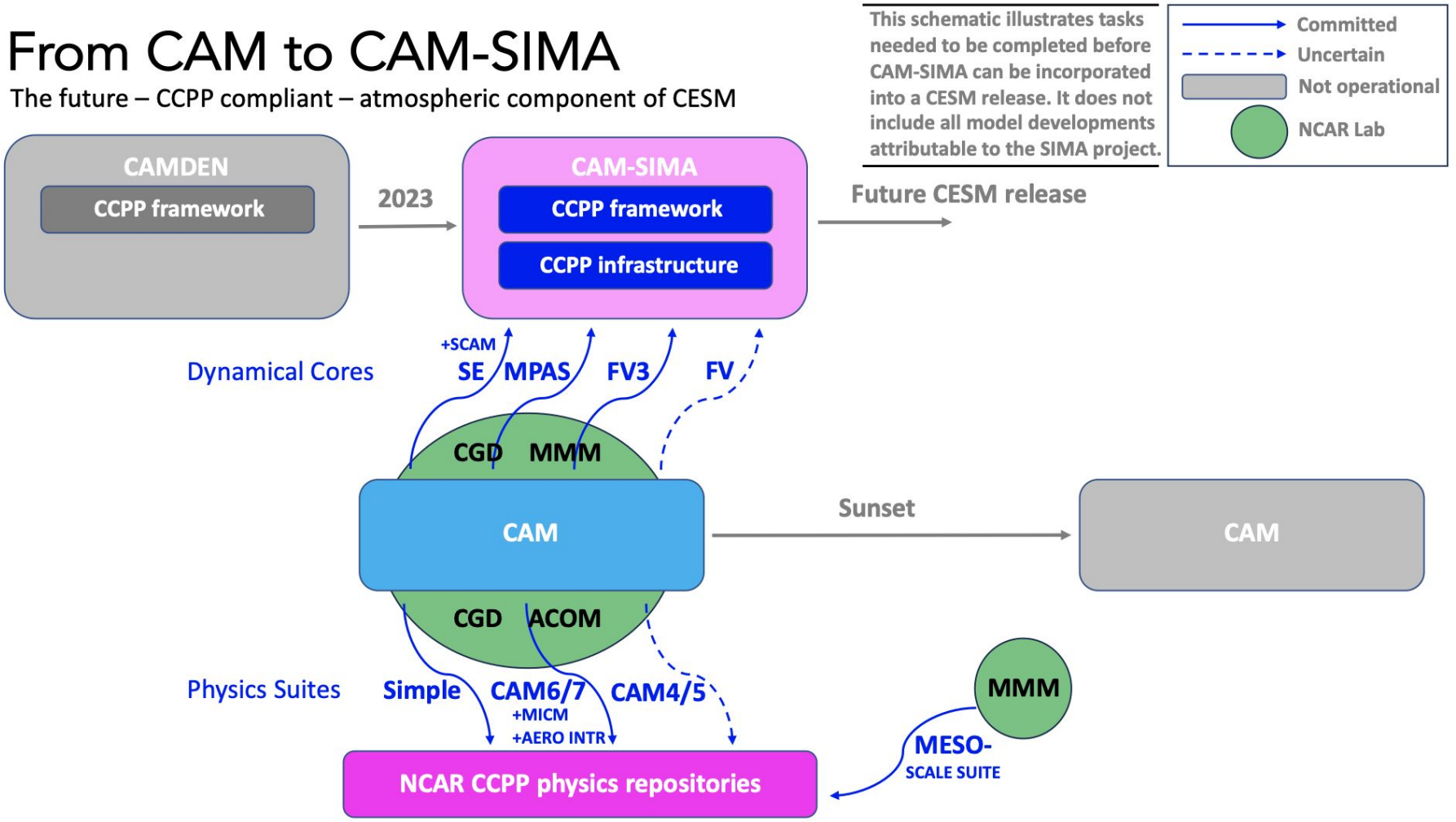
1	Snapshot Variable Name	Snapshot Longname	Longname Location	Snapshot Units	CCPP Standard Name	Accepted
400	ptend_taux_top	net zonal stress at top of model	physics_types.F90: line 163	Pa	N/A	
401	ptend_tauy_srf	net meridional stress at surface	physics_types.F90: line 164	Pa	N/A	
402	ptend_tauy_top	net meridional stress at top of model	physics_types.F90: line 165	Pa	N/A	
403					TPHYSAC VARIABLES	
404	tphysac_cmfmc	convective mass flux (m sub c)	physpkg.F90: line 2072		atmosphere_convective_mass_flux_due_to_all_convection	X
405	tphysac_det_ice	vertical integral of detrained ice	physpkg.F90: line 2141		vertically_integrated_detrainment_of_ice_due_to_all_convection	X
406	tphysac_det_s	vertical integral of detrained static energy from ice	physpkg.F90: line 2140		vertically_integrated_heating_from_freezing_of_detrained_liquid_due_to_all_convection	X
407	tphysac_dlf	detraining cld H2O from shallow + deep convections	physpkg.F90: line 2076		detrainment_of_water_due_to_all_convection	X
408	tphysac_dlf2	detraining cld H2O from shallow convections	physpkg.F90: line 2077		detrainment_of_water_due_to_shallow_convection	X
409	tphysac_fh2o	h2o flux to balance source from methane chemistry	physpkg.F90: line 1433		vertically_integrated_water_flux_due_to_chemistry	X
410	tphysac_fix_heat	heat flux for check_energy_chng	physpkg.F90: line 1434			
411	tphysac_net_flx		physpkg.F90: line 2069		net_radiative_fluxes_through_top_and_bottom_of_atmosphere_column	X
412	tphsyac_obklen	Obu[illegible]	[illegible]		[illegible]	X
413	tphysac_rliq	vert			loop	X
414	tphysac_rliq2	vert			time_loop	X
415	tphysac_surfric	surf				X
416	tphysac_zdu	detr				X
417						
418	tphysbc_cmfcme	cmf				X
419	tphysbc_cmfmc	con				X
420	tphysbc_dlf	detr				X
421	tphysbc_dlf2	detr				X
422	tphysbc_fix_heat	hea			ce_upward_heat_flux_in_air, surfa	
423	tphysbc_net_flx					X
424	tphysbc_pflx	conv rain flux throughout bottom of lev	physpkg.F90: line 2078		precipitation_flux_at_interface_due_to_deep_convection	X
425	tphysbc_rice	vertical integral of ice not yet in q (ixcldice)	physpkg.F90: line 2138		vertically_integrated_cloud_ice_tendency_due_to_all_convection_to_be_applied_later_in_time_loop	X
426	tphysbc_rliq	vertical integral of liquid not yet in q (ixcldliq)	physpkg.F90: line 2137		vertically_integrated_cloud_liquid_tendency_due_to_all_convection_to_be_applied_later_in_time_loop	X
427	tphysbc_rliq2	vertical integral of liquid from shallow scheme	physpkg.F90: line 2139		vertically_integrated_cloud_liquid_tendency_due_to_shallow_convection_to_be_applied_later_in_time_loop	X
428	tphysbc_zdu	detraining mass flux from deep convection	physpkg.F90: line 2071		detrainment_mass_flux_due_to_deep_convection	X
429						
430					NCAR/ATMOSPHERIC_PHYSICS	
431	**Variable Name**	**CAM equivalent variable**	**Meta File**		**CCPP Standard Name**	
432	rho	No CAM equivalent	kessler.meta		density_of_dry_air	X
433	qr	state%q(:,:,ixrain)	kessler.meta		rain_mixing_ratio_wrt_dry_air	X
434	scheme_name	No CAM equivalent	kessler.meta		scheme_name	X
435	rair	rairv	geopotential_t.meta		composition_dependent_gas_constant_of_dry_air	X
436	zvir	zvirv	geopotential_t.meta		ratio_of_water_vapor_gas_constant_to_composition_dependent_dry_air_gas_constant_minus_one	X
437	zi	state%zi	geopotential_t.meta		geopotential_height_wrt_surface_at_interface	X
438	dudt	ptend%u	physics_tendency_updaters.meta		tendency_of_eastward_wind	
439	dudt_total	tend%u	physics_tendency_updaters.meta		tendency_of_eastward_wind_due_to_model_physics	
440	dvdt	ptend%v	physics_tendency_updaters.meta		tendency_of_northward_wind	

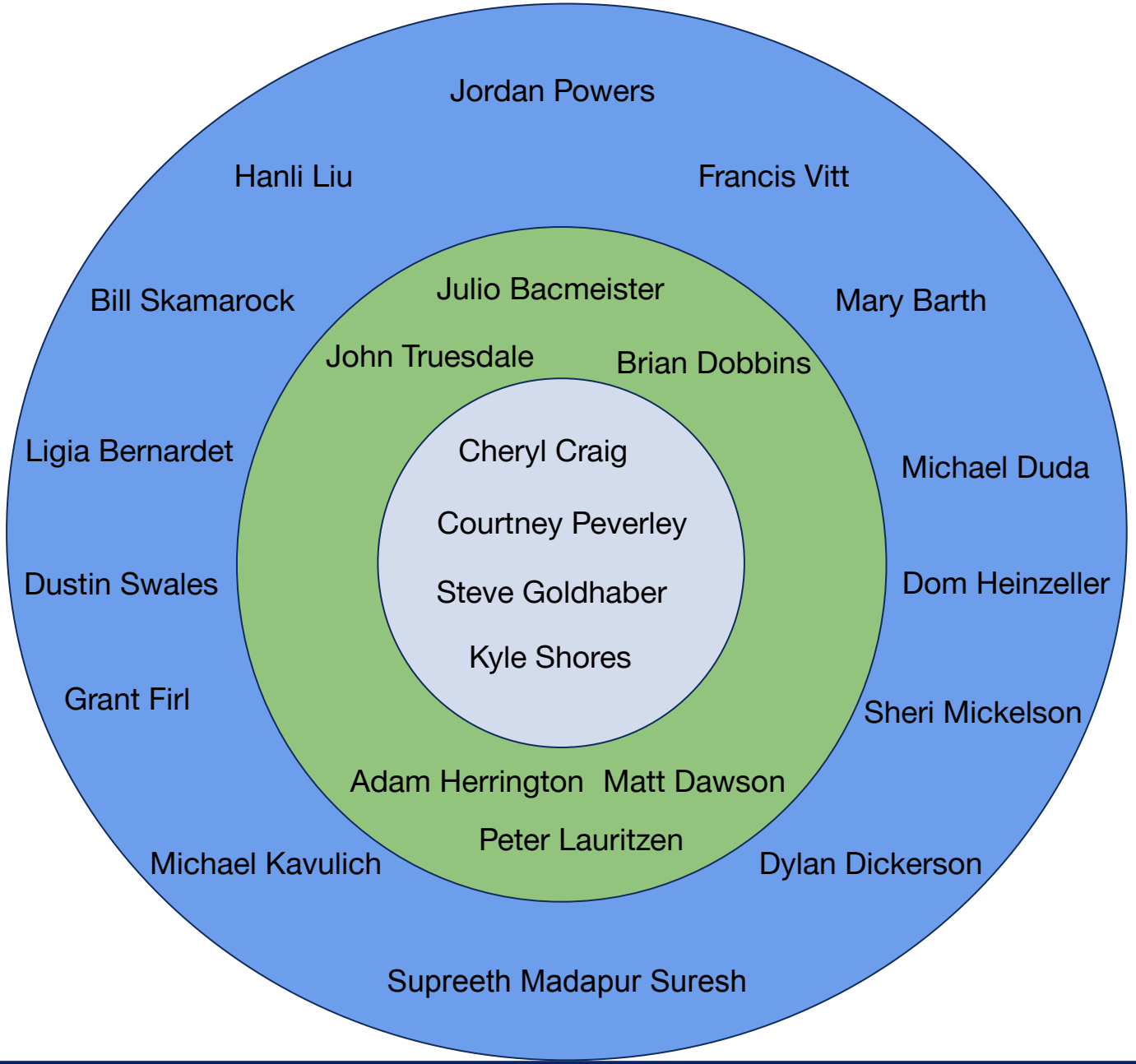
It is slow, tedious work, but has already found both **software bugs** (e.g. variables that aren't being used or are duplicated) and **science issues** (e.g. variables that didn't represent the physical quantities the developer thought they did).

CCPPStandardNames (repo #4)



Summary


From CAM to CAM-SIMA
The future – CCPP compliant – atmospheric component of CESM
This schematic illustrates tasks needed to be completed before CAM-SIMA can be incorporated into a CESM release. It does not include all model developments attributable to the SIMA project.
Committed
Uncertain
Not operational
NCAR Lab
CAMDEN
CCPP framework
2023
CAM-SIMA
CCPP framework
CCPP infrastructure
Future CESM release
Dynamical Cores
+SCAM
SE
MPAS
FV3
FV
CGD
MMM
CAM
Sunset
CAM
CGD
ACOM
Physics Suites
Simple
CAM6/7
+MICM
+AERO INTR
CAM4/5
MMM
MESO-
SCALE SUITE
NCAR CCPP physics repositories


Jordan Powers
Hanli Liu
Francis Vitt
Bill Skamarock
Julio Bacmeister
Mary Barth
John Truesdale
Brian Dobbins
Ligia Bernardet
Cheryl Craig
Michael Duda
Courtney Peverley
Dustin Swales
Steve Goldhaber
Dom Heinzeller
Kyle Shores
Grant Firl
Sheri Mickelson
Adam Herrington
Matt Dawson
Peter Lauritzen
Michael Kavulich
Dylan Dickerson
Supreeth Madapur Suresh

https://dtcenter.org/events/2023/ccpp-visioning-workshop

2023 CCPP Visioning Workshop (virtual, week of August 14, 2023)

What: discuss future direction for CCPP

Who

- CCPP project leads and developers
- Physics and atmospheric composition/chemistry developers
- Scientists working on coupling for ESMs, especially physics-dynamics coupling
- NOAA, NCAR, NRL, NASA, academia, private sector, etc.

Which models: UFS, SCM, NEPTUNE, SIMA etc.

Goals

- Inform the community about capabilities
- Gather input from developers/SMEs
- Discuss best practices for interoperability and collaborative development
- Create a prioritized list of required advancements for CCPP

Desired outcome

- Common understanding of the state of CCPP and prioritized requirements and needs to meet scientific and technological frontiers in the next 5-10 years

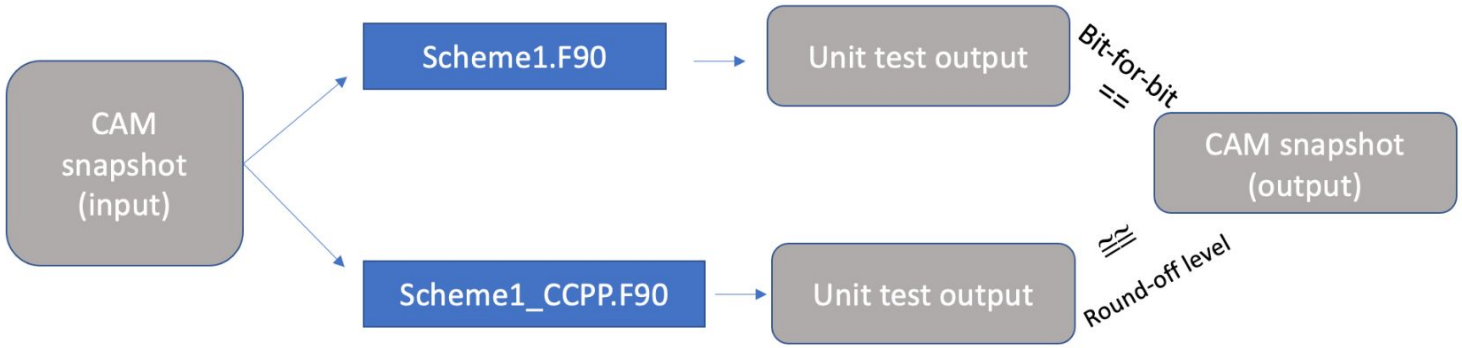
Questions?

Thanks for listening!

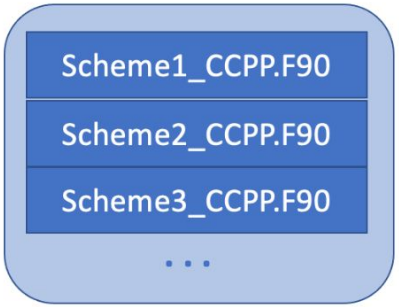
CCPP Implementation plan

Refactoring)

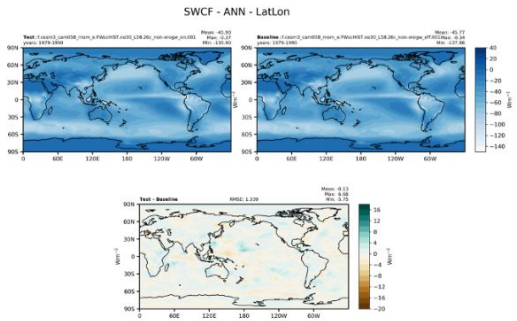
Validation)



Suite Generation)



Science Validation)



Challenges

There are several challenges that we are facing for this work:

1. Time/people -> Making sure there are enough SEs that can move this work forward without being pulled off onto other tasks/duties.

1. Technical -> Need to make sure the framework has the capacity to deal with the various edge/corner cases that can exist in CAM's physics schemes.