

EAMxx infrastructure

Designing a next generation global
atmosphere model

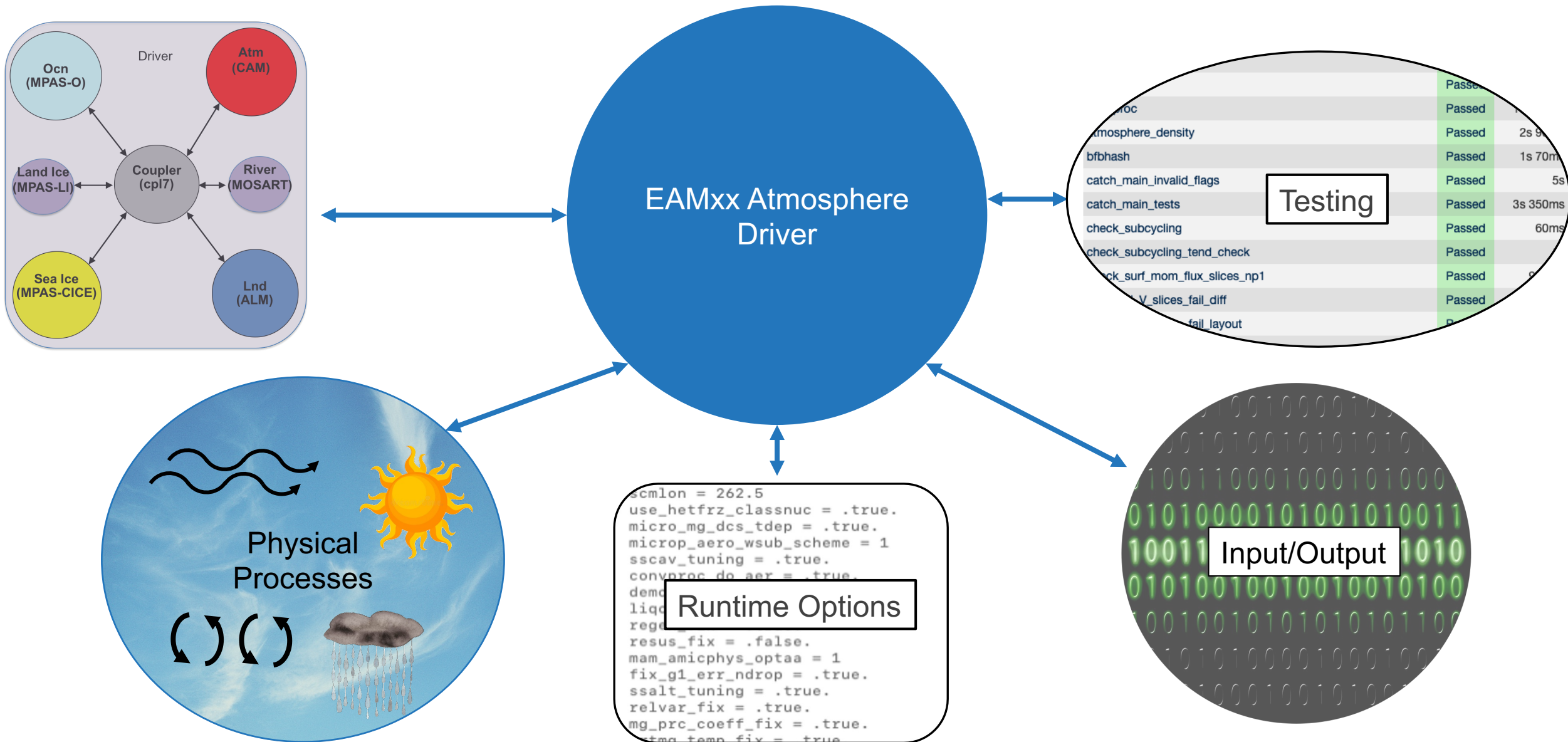
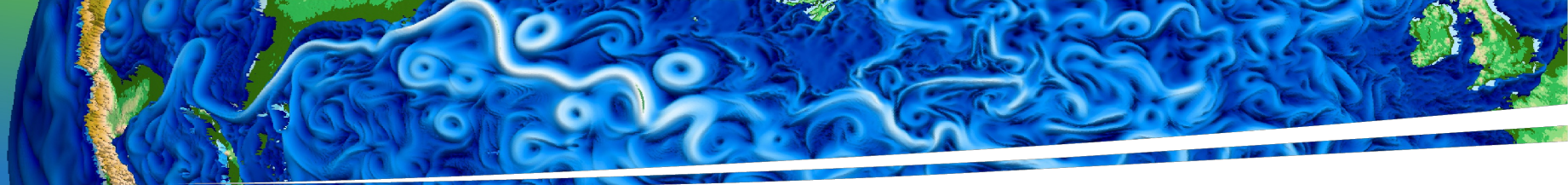
Aaron S. Donahue

E3SM All-Hands, April 11, 2024



This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 IM Release Number LLNL-PRES-862664

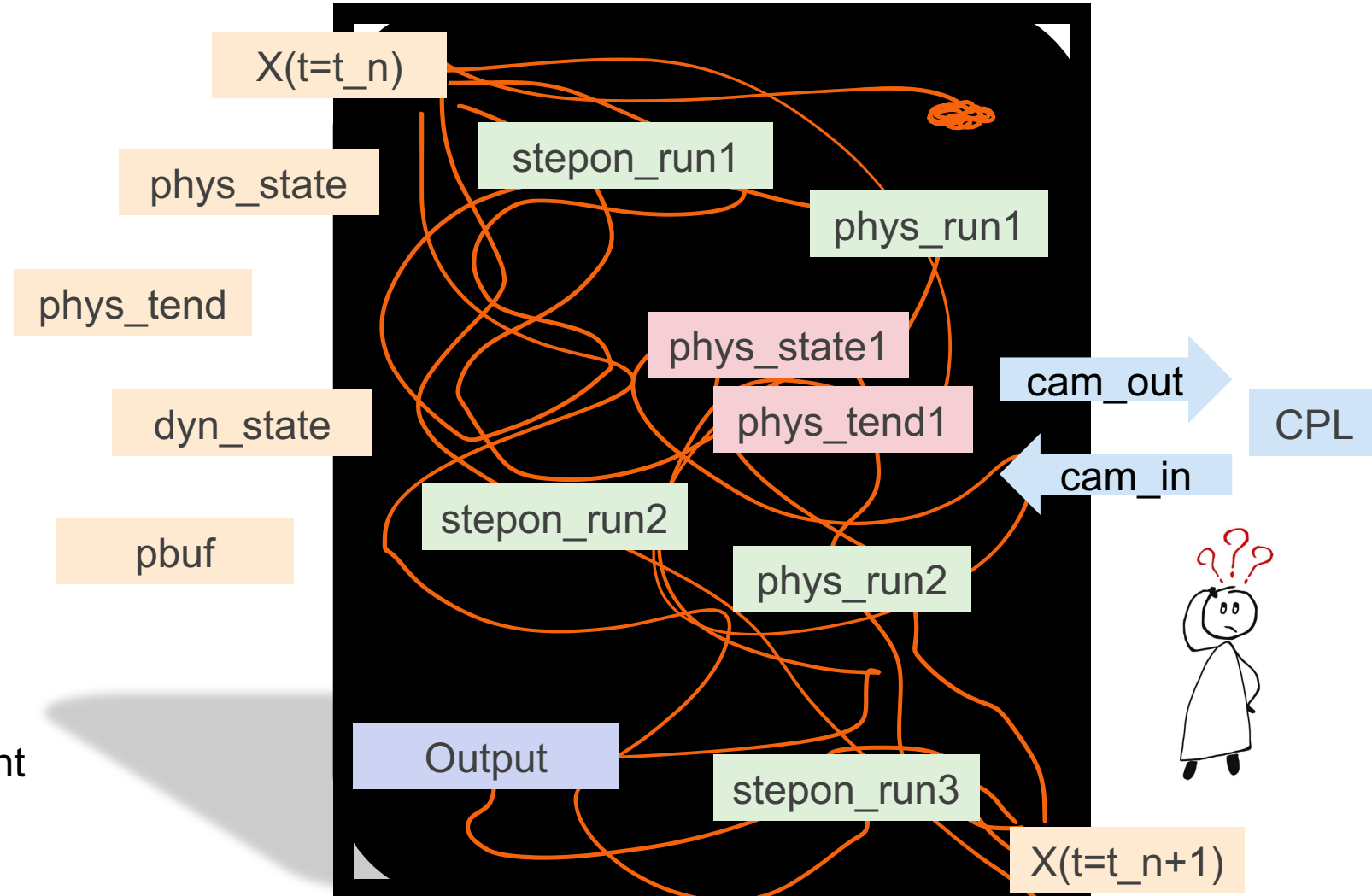


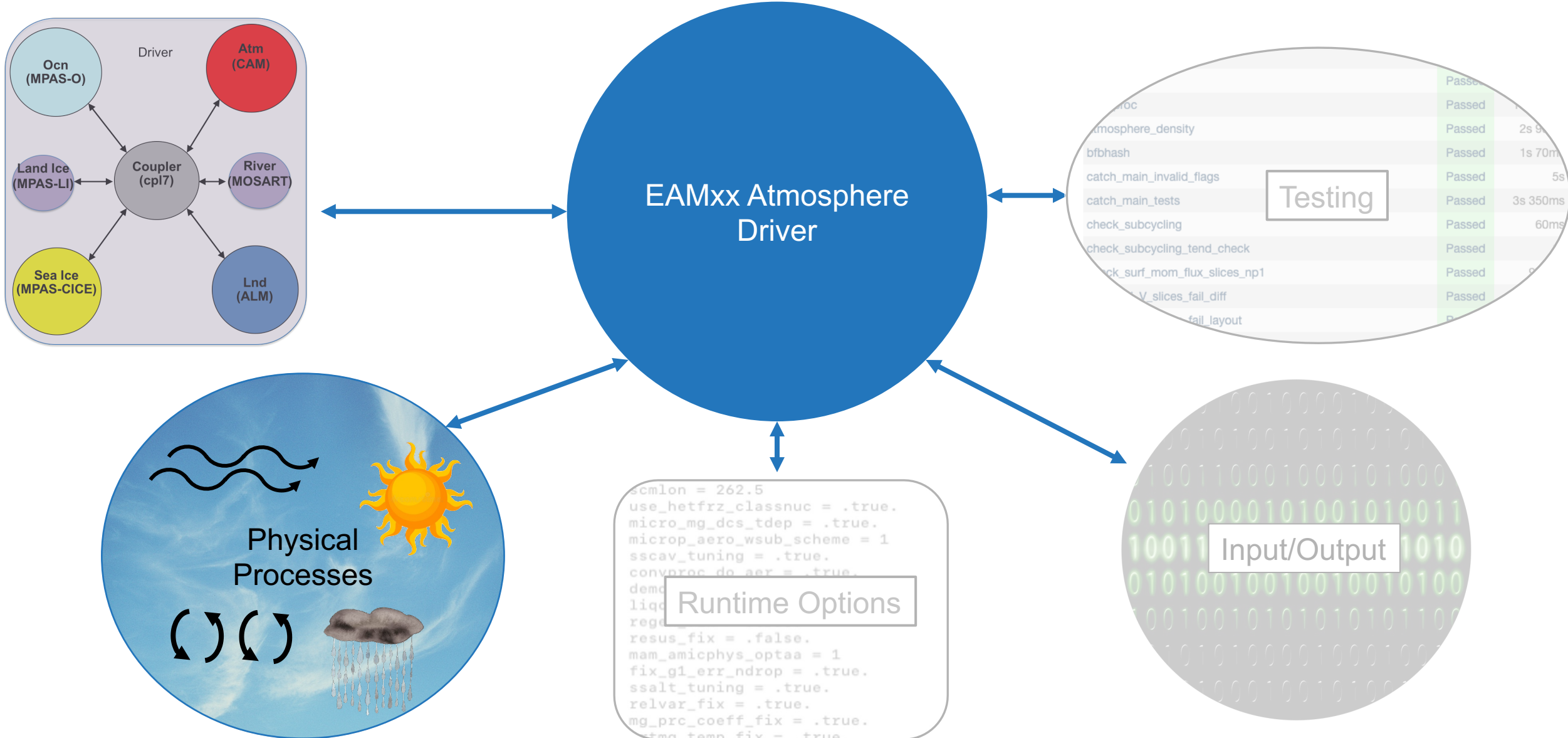


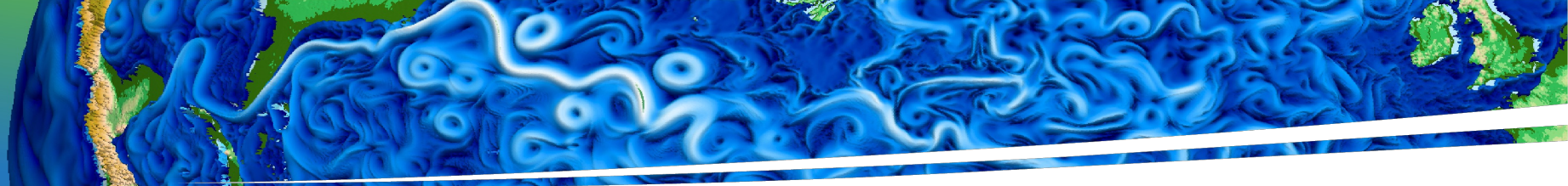


Current EAM paradigm

- Opaque
- Hardcoded process order,
- Hardcoded time-splitting,
- Difficult to add new processes,
- Difficult to audit
- Individual pieces are dependent on the rest of the model.







The new EAMxx paradigm

phys_state
phys_tend
dyn_state
PBUF
cam_in/out



Field Manager Class and **Field Objects**

- **All persistent variables** are instances of a field object
- All fields managed by a **single Field Manager**
- Simplifies operations on fields, e.g.
 - Remapping to different grids
 - Adding new fields
 - Storing field metadata

cam_init
phys_init
dyn/stepon_init
cam_run,1,2,3,4
phys_run1,2
stepon_run1,2,3



What is a Field?

- Stores everything the atmosphere model needs to know about a global variable.
- As an object allows SCREAM to define universal operations for all fields,
 - E.g. remapping, arithmetic operations, cloning, property checking, timestep management ...

- > Name: T_mid
- > Units: K
- > Layout: (columns, levels)
- > Grid: Physics PG2
- > Datatype: Real
- > Timestamp: TS{2024,11,4,8,30,0}
- > Data Pointer: 6efgkh38sahdlgis0372
- > Providers: HOMME, SHOC, P3, RRTMGP, Surface-Coupling
- > Customers: HOMME, SHOC, P3, RRTMGP, Surface-Coupling
- > Extra Metadata: [Mask, Foo, Bar, ...]

* Artist depiction



The new EAMxx paradigm

phys_state
phys_tend
dyn_state
PBUF
cam_in/out



Field Manager Class and **Field Objects**

- **All persistent variables** are instances of a field object
- All fields managed by a **single Field Manager**
- Simplifies operations on fields, e.g.
 - Remapping to different grids
 - Adding new fields
 - Storing field metadata

cam_init
phys_init
dyn/stepon_init
cam_run,1,2,3,4
phys_run1,2
stepon_run1,2,3



Atmosphere Driver Class and Atmosphere Process Objects

- Every process (**physics and dynamics**) is an instance of an AP object.
- **Atmosphere Driver manages all AP's**, including call sequence order, subcycling, grids, IO
- Able to handle **many runtime configurations**.

What is an Atmosphere Process?

- An interface to the dynamics or an atmosphere parameterization.
- EAMxx supports universal operations on atmosphere processes:
 - Unit testing, subcycling, backing out tendencies, log msgs, performance timing, ...
- Has a defined initialization, run and finalization.

```
> Name: SHOC
> Type: Physics
> Grid: Physics PG2
> Timestamp: TS{2024,11,4,8,30,0}
> Inputs: omega, surf_sens_flux, surf_mom_flux, ..., T_mid, qv
> Outputs: surf_evap, T_mid, qv, ... , tke, pbl_height
> init_impl: Do 'xyz' to initialize SHOC
> run_impl: Do 'ijk,' then call shoc_main, do 'qrs'
> finalize_impl: Nothing to do
```

* Artist depiction



The new EAMxx paradigm

```
add_field<X>(...)
```

X:

```
Required (IN)
Computed (OUT)
Updated (IN/OUT)
```



Field Manager Class and **Field Objects**

- **All persistent variables** are instances of a field object
- All fields managed by a **single Field Manager**
- Simplifies operations on fields, e.g.
 - Remapping to different grids
 - Adding new fields
 - Storing field metadata

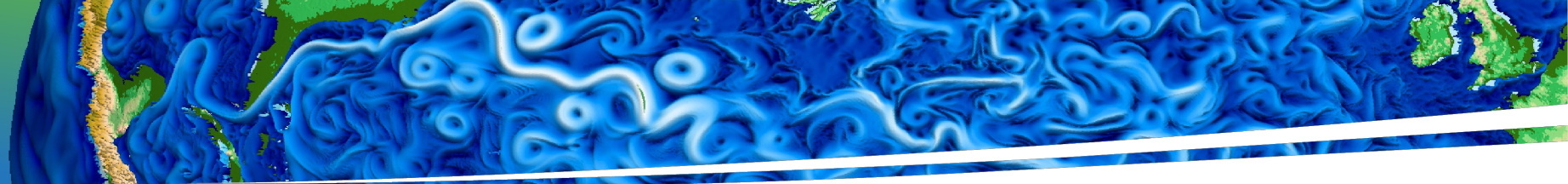
```
ad.initialize(comm,params,t0)
ad.run(dt)
ad.finalize()
```

```
ap.initialize(t0,type)
ap.run(dt)
ap.finalize()
```

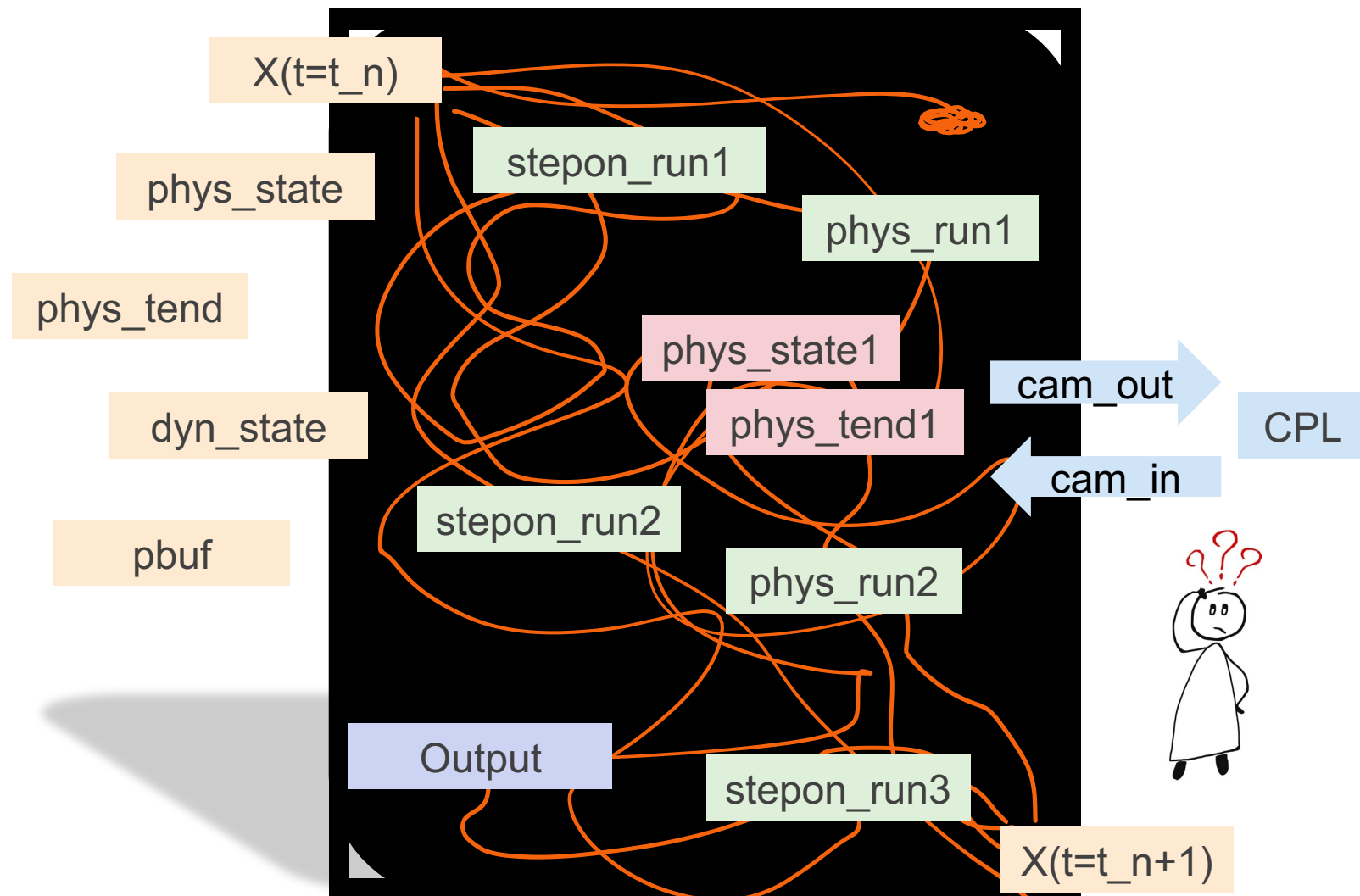


Atmosphere Driver Class and Atmosphere Process Objects

- Every process (**physics and dynamics**) is an instance of an AP object.
- **Atmosphere Driver manages all AP's**, including call sequence order, subcycling, grids, IO
- Able to handle **many runtime configurations**.



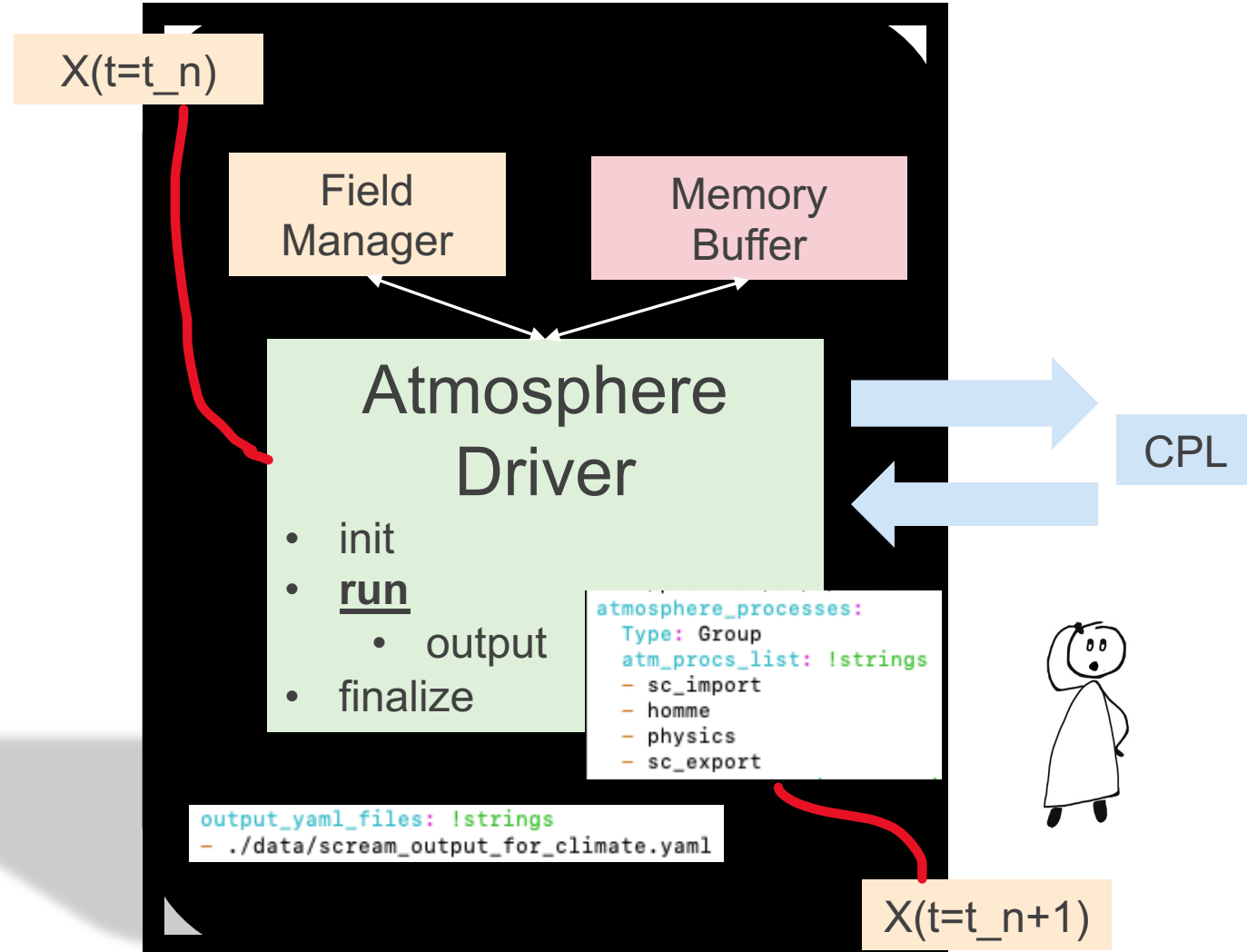
Current EAM paradigm





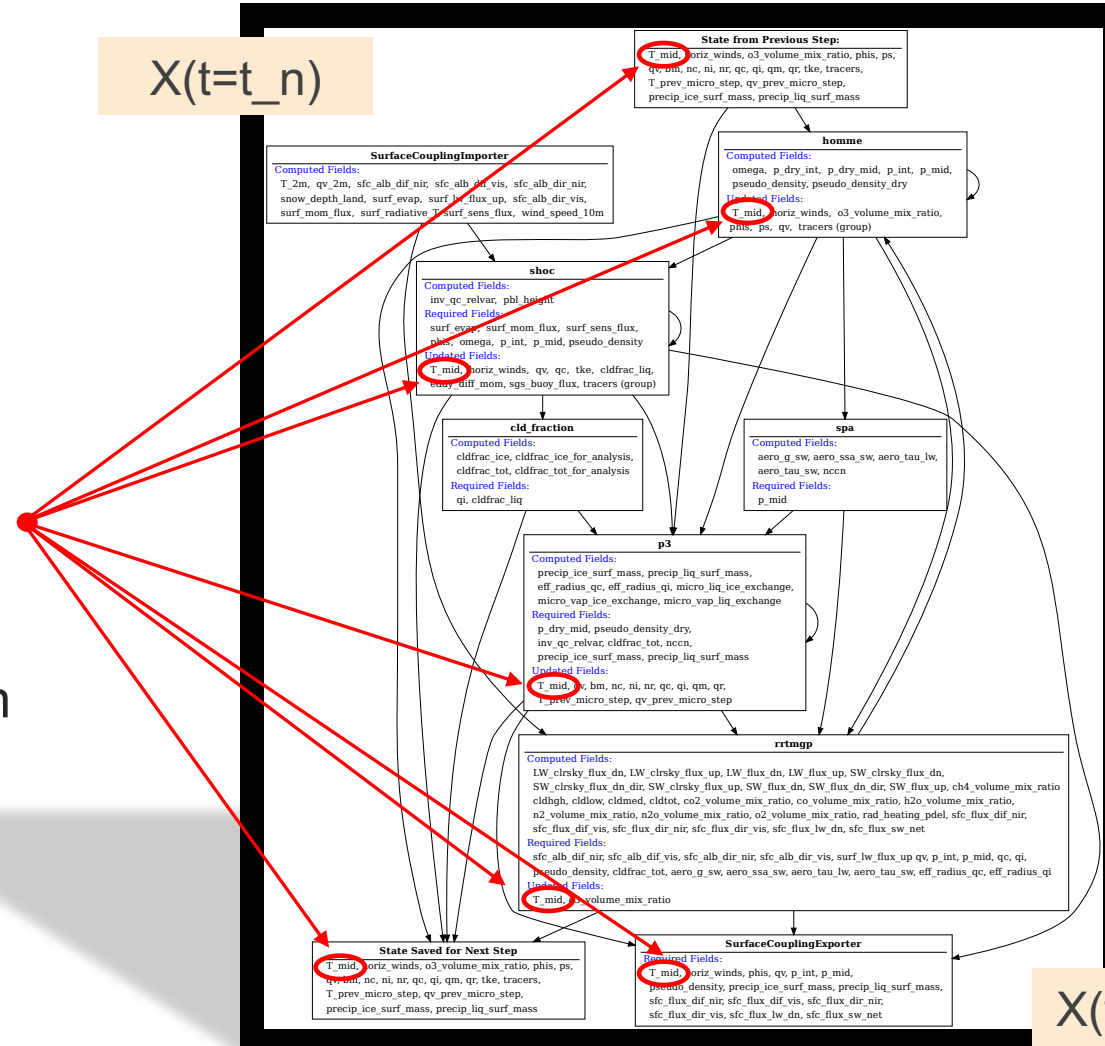
The EAMxx paradigm

- Simpler
- Increased runtime options, and process ordering set at runtime.
- Interoperability for atmosphere processes.
- Independence for individual processes.

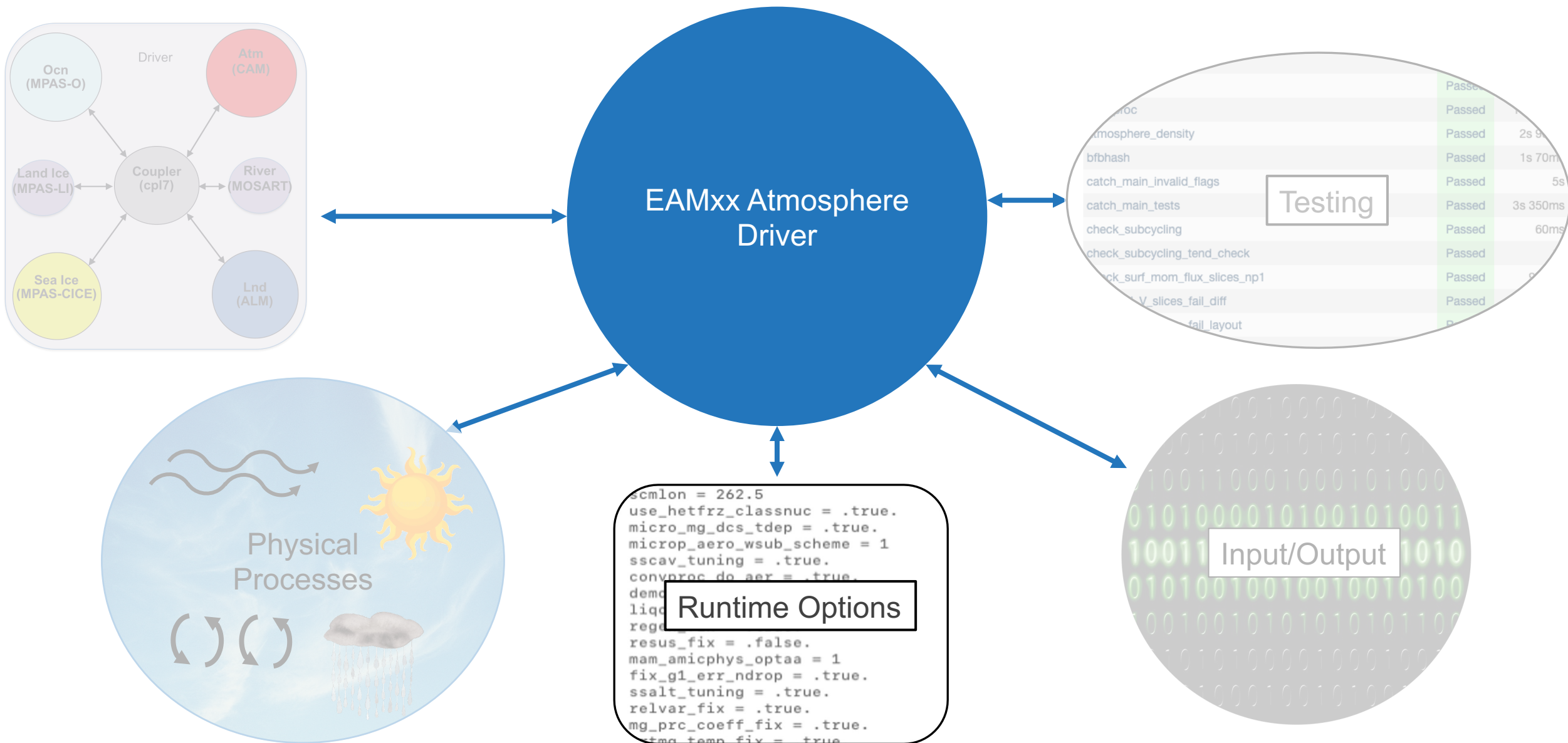
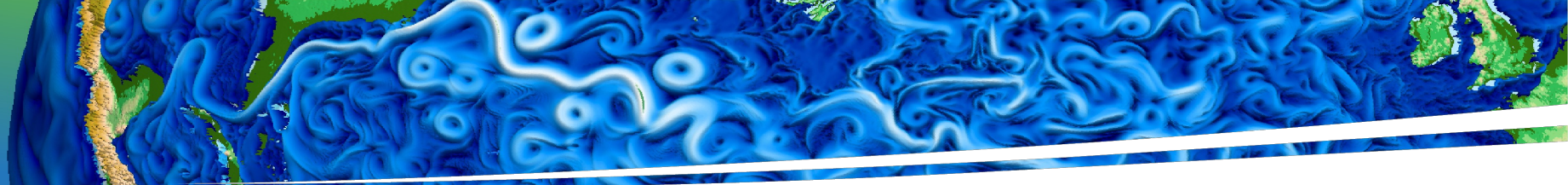


Directed Acyclic Graph (DAG)

- Reference for **how** state variables are used.
- Quickly **audit** all global variables in the simulation.
- Incredibly useful for **debugging**.



X(t=t_n+1)



Runtime Options: namelist

```
cld_macmic_num_steps = 6
mfilt = 10000,1,12
cosp_lite = .true.
use_gw_front = .true.
nhtfrq = 1,0,-2
fincl1 = 'Q', 'CLDLIQ', 'CLDICE'
fincl2 = 'PS', 'TMQ', 'T500'
scm_iop_srf_prop = .true.
avgflag_pertape = 'A'
iop_nudge_tq = .false.
cld_macmic_num_steps
iradlw = 1
iradsw = 1
iopfile = '/usr/gdata
precip_off = .false.
scmlat = 36.6
scmlon = 262.5
use_hetfrz_classnuc =
micro_mg_dcs_tdep = .
microp_aero_wsub_sche
sscav_tuning = .true.
convproc_do_aer = .tr
demott_ice_nuc = .tru
liqcf_fix = .true.
regen_fix = .true.
resus_fix = .false.
mam_amicphys_optaa =
fix_g1_err_ndrop = .t
ssalt_tuning = .true.
relvar_fix = .true.
mg_prc_coeff_fix = .true.
rrtmg_temp_fix = .true.
mam_amicphys_optaa = 1
fix_g1_err_ndrop = .true.
ssalt_tuning = .true.
use_rad_dt_cosz = .true.
ice_sed_ai = 500.0
do_tms = .false.
n_so4_monolayers_pcase = 8.0D0
se_ftype = 2
zmconv_trigdcape_ull = .true.
cld_sed = 1.0D0
effgw_beres = 0.35
gw_convect_hcf = 12.5
effaw_oro = 0.375
```

- Difficult to read/parse
- Not organized
- Cumbersome to add new options
- Difficult to audit, vulnerable to user error
 - Conflicts only detected if developer adds their own checks.
 - Multiple entries can lead to unexpected behavior.



Runtime Options: YAML

```

> ./atmquery mac_aero_mic::number_of_subcycles
namelist_defaults::atmosphere_processes::physics::mac_aero_mic::number_of_subcycles: 6
> ./atmchange mac_aero_mic::number_of_subcycles=12
Regenerating .../namelist_scream.xml. Manual edits will be lost.
> ./atmquery mac_aero_mic::number_of_subcycles
namelist_defaults::atmosphere_processes::physics::mac_aero_mic::number_of_subcycles: 12
  
```

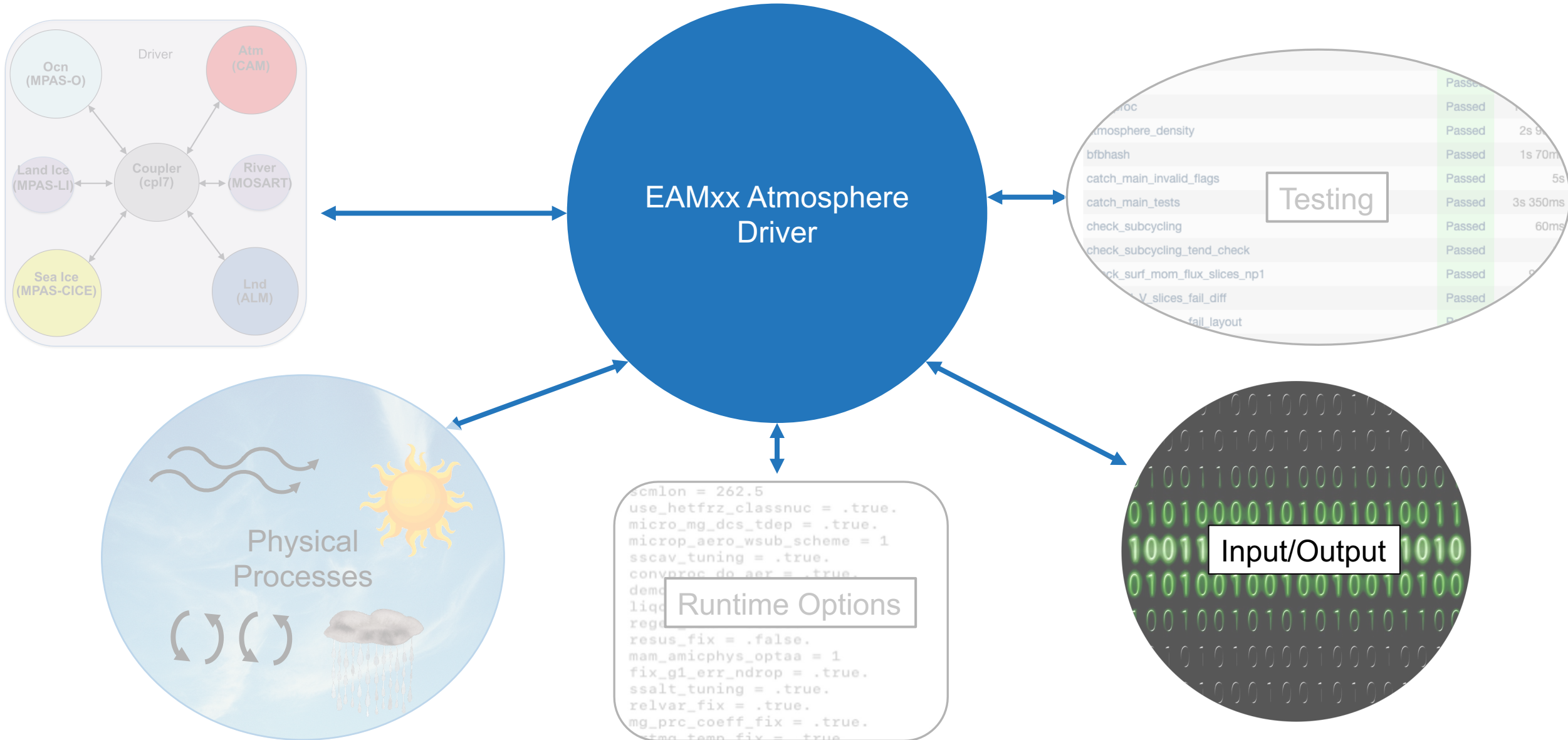
- EAMx
- Human section
- Strict
- Support
- Setting procedure
- Has s
 - ./
 - ./

```

mac_aero_mic:
  Type: Group
  atm_procs_list: !strings
  - tms
  - shoc
  - cldFraction
  - spa
  - p3
  cldFraction:
    compute_tendencies: !strings []
    enable_postcondition_checks: true
    enable_precondition_checks: true
    internal_diagnostics_level: 0
    number_of_subcycles: 1
    repair_log_level: trace
  compute_tendencies: !strings []
  enable_postcondition_checks: true
  enable_precondition_checks: true
  internal_diagnostics_level: 0
  number_of_subcycles: 12
  
```

```

- horiz_winds
enable_postcondition_checks: true
enable_precondition_checks: true
internal_diagnostics_level: 0
mac_aero_mic:
  Type: Group
  atm_procs_list: !strings
  - tms
  - shoc
  - cldFraction
  - spa
  - p3
  cldFraction:
    compute_tendencies: !strings []
    enable_postcondition_checks: true
    enable_precondition_checks: true
    internal_diagnostics_level: 0
    number_of_subcycles: 1
    repair_log_level: trace
  compute_tendencies: !strings []
  enable_postcondition_checks: true
  enable_precondition_checks: true
  internal_diagnostics_level: 0
  number_of_subcycles: 12
p3:
  compute_tendencies: !strings []
  do_predict_ccn: true
  do_prescribed_ccn: true
  enable_column_conservation_checks: false
  enable_postcondition_checks: true
  enable_precondition_checks: true
  internal_diagnostics_level: 0
  max_total_ni: 740000.0
  number_of_subcycles: 1
  p3_a_imm: 0.65
  p3_autoconversion_prefactor: 1350.0
  
```





Output Control: user_nl_eam

```
cld_macmic_num_steps = 6  
mfilt = 10000,1,12  
cosp_lite = .true.  
use_gw_front = .true.  
nhtfrq = 1,0,-2  
fincl1 = 'Q','CLDLIQ','CLDICE'  
fincl2 = 'PS','TMQ','T500'  
scm_iop_srf_prop = .true.  
avgflag_pertape = 'A','I','X'  
iop_nudge_ta = false
```

Output Control: YAML

```
cld_macmic_num_steps = 6
mfilt = 10000,1,12
cosp_lite = .true.
use_gw_front = .true.
nhtfrq = 1,0,-2
fincl1 = 'Q','CLDLIQ','CLDICE'
fincl2 = 'PS','TMQ','T500'
scm_iop_srf_prop = .true.
avgflag_pertape = 'A','I','X'
iop_nudge_to = false
```

```
Scorio:
  model_restart:
    filename_prefix: ./F2010-SCREAMv1.ne30pg2_ne30pg2
  output_control:
    Frequency: 1
    frequency_units: nmonths
  output_yaml_files: !strings
  - data/my_first_output_file.yaml
  - data/my_other_output_file.yaml
```

Output Control: YAML

```

cld_macmic_num_steps = 6
mfilt = 10000,1,12
cosp_lite = .true.
use_gw_front = .true.
nhtfrq = 1,0,-2
fincl1 = 'Q','CLDLIQ','CLDICE'
fincl2 = 'PS','TMQ','T500'
scm_iop_srf_prop = .true.
avgflag_pertape = 'A','I','X'
iop_nudge_to = false

```

```

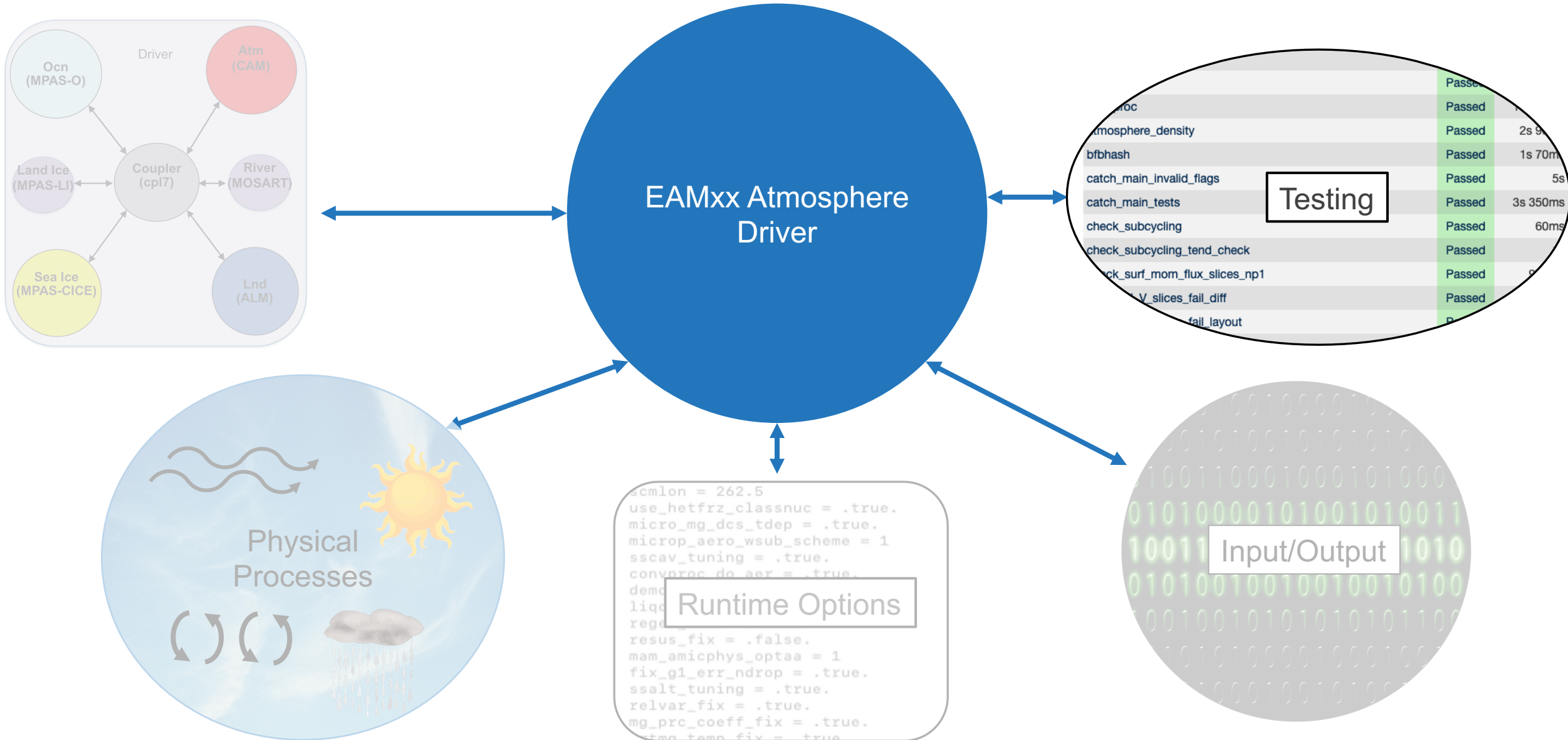
Averaging Type: Instant
Fields:
  Physics PG2:
    Field Names:
      - qv
      - qc
      - qi
Max Snapshots Per File: 1
filename_prefix: output.scream.monthly.NAME_1
output_control:
  Frequency: 1
  frequency_units: nmonths

```

```

Averaging Type: Max
Fields:
  Physics PG2:
    Field Names:
      - ps
      - VapWaterPath
      - LiqWaterPath
Max Snapshots Per File: 12
filename_prefix: output.scream.monthly.NAME_2
output_control:
  Frequency: 2
  frequency_units: nhours


```





Testing:

- EAMxx inherits all the standard E3SM testing coverage, + **robust unit tests**
- This is possible because,
 - Model elements are **independent**.
 - Take advantage of arbitrary class structure in C++.
- Incredibly useful for **debugging** and **verification**
- SCREAM git-repo CI integrates unit testing through the Autotester.




E3SM-Autotester commented 6 hours ago Member ...

Status Flag 'Pull Request AutoTester' - Jenkins Testing: 1 or more Jobs FAILED

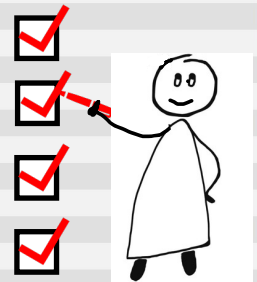
Note: Testing will normally be attempted again in approx. 2 Hrs. If a change to the PR source branch occurs, the testing will be attempted again on next available autotester run.

- ▶ Pull Request Auto Testing has FAILED (click to expand)
- ▶ SCREAM_PullRequest_Autotester_Mappy # 5172 FAILED (click to see last 100 lines of console output)
- ▶ SCREAM_PullRequest_Autotester_Weaver # 5520 FAILED (click to see last 100 lines of console output)



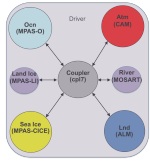
477 tests passed.

| Name ^ | Status ^ | Time | History | Summary |
|--------------------------------|----------|----------|---------|---------|
| ad_ut | Passed | 2s 140ms | Stable | Stable |
| array_io | Passed | 3s 230ms | Stable | Stable |
| atm_proc | Passed | 2s 680ms | Stable | Stable |
| atmosphere_density | Passed | 1s 460ms | Stable | Stable |
| bfbhash | Passed | 3s 730ms | Stable | Stable |
| catch_main_invalid_flags | Passed | 3s 990ms | Stable | Stable |
| catch_main_tests | Passed | 3s 640ms | Stable | Stable |
| check_subcycling | Passed | 50ms | Stable | Stable |
| check_subcycling_tend_check | Passed | 320ms | Stable | Stable |
| check_surf_mom_flux_slices_np1 | Passed | 370ms | Stable | Stable |
| check_surf_mom_flux_slices_np2 | Passed | 380ms | Stable | Stable |
| check_surf_mom_flux_slices_np3 | Passed | 340ms | Stable | Stable |
| check_surf_mom_flux_slices_np4 | Passed | 370ms | Stable | Stable |
| check_U_V_slices_fail_diff | Passed | 310ms | Stable | Stable |
| check_U_V_slices_fail_layout | Passed | 390ms | Stable | Stable |
| check_U_V_slices_fail_missing | Passed | 350ms | Stable | Stable |
| check_U_V_slices_np1 | Passed | 470ms | Stable | Stable |
| check_U_V_slices_np2 | Passed | 360ms | Stable | Stable |
| check_U_V_slices_np3 | Passed | 370ms | Stable | Stable |
| check_U_V_slices_np4 | Passed | 340ms | Stable | Stable |
| cid_fraction_standalone | Passed | 1s 570ms | Stable | Stable |
| coarsening_remapper_np1 | Passed | 3s 400ms | Stable | Stable |
| coarsening_remapper_np2 | Passed | 2s 620ms | Stable | Stable |
| coarsening_remapper_np3 | Passed | 1s 180ms | Stable | Stable |
| coarsening_remapper_np4 | Passed | 820ms | Stable | Stable |





Conclusions:



- EAMxx is a complete rewrite of the E3SM global atmosphere model infrastructure in C++/Kokkos



- EAMxx incorporates modern software best practices

```

cmion = 262.5
use_netfz_classnuc = .true.
micro_mg_dcs_tdep = .true.
micro_aero_waub_scheme = 1
scaav_tuning = .true.
convproc_d0_aer = .true.
demott_ice_nuc = .true.
liqcr_fix = .true.
regen_fix = .true.
reus_fix = .false.
mas_antiophys_optaa = 1
fix_gl_err_ndrop = .true.
small_tuning = .true.
relvar_fix = .true.
q_prec_coeff_fix = .true.
  
```

- The adoption of YAML makes runtime options:
 - More protected against user error,
 - Human readable,
 - More organized

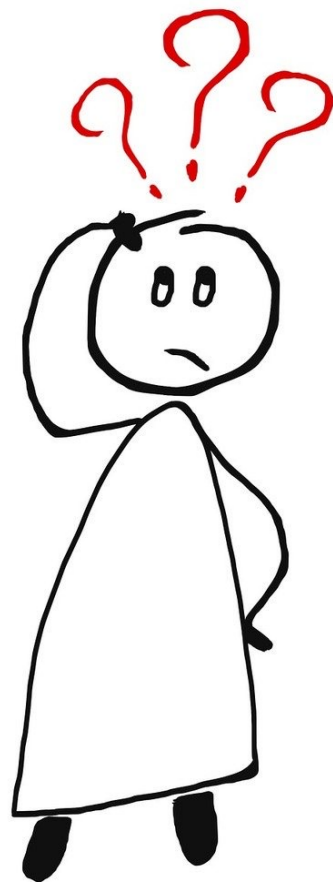
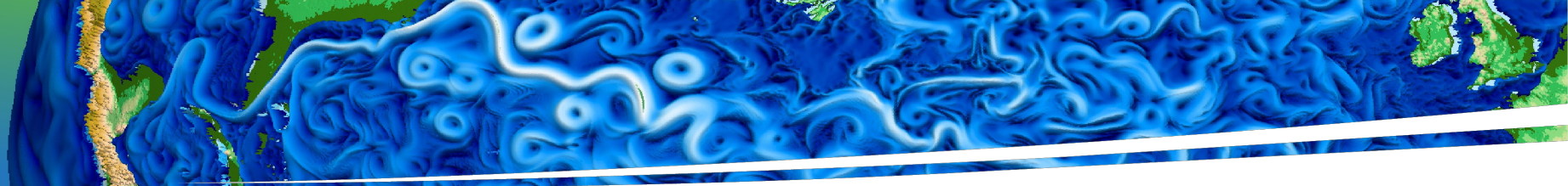


- The benefits of YAML runtime options is extended to output control.



| Test Name | Result | Time |
|-----------------------------|--------|----------|
| atmosphere_density | Passed | 21s |
| atmosphere | Passed | 11 170s |
| catch_map_tests | Passed | 5s |
| catch_map_tests | Passed | 2s 200ms |
| check_sweeping | Passed | 60ms |
| check_subcycling_land_check | Passed | |
| x_surf_mom_flux_albac_opt1 | Passed | |
| gl_albac_opt1 | Passed | |
| gl_albac_opt2 | Passed | |

- EAMxx infrastructure was designed with testing in mind. Unit testing:
 - Makes debugging substantially easier,
 - Gives more confidence that new features won't impact or break existing features.



e3sm.org



Performance Portability (C++/Kokkos)

U.S. DEPARTMENT OF **ENERGY** | Office of Science

Sandia National Laboratories **E³SM** Energy Exascale Earth System Model

SC23 Denver, CO | am hpc.

The Simple Cloud-Resolving E3SM Atmosphere Model Running on the Frontier Exascale System

Luca Bertagna (lbtag@sandia.gov)
Sandia National Laboratories
Nov. 15th 2023

Mark Taylor, Peter Caldwell, Luca Bertagna, Conrad Clevenger, Aaron Donahue, James Foucar, Oksana Guba, Benjamin Hillman, Noel Keen, Jayesh Krishna, Matthew Norman, Sarat Sreepathi, Christopher Terai, James White, Danqing Wu, Andrew Salinger, Renata McCoy, L. Ruby Leung, David Bader




Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.


<https://acme-climate.atlassian.net/wiki/spaces/ECM/pages/4129325057/2024-02-15+All-Hands+Presentation+Meeting+Notes>



Performance Portability (C++/Kokkos)

Packs


U.S. DEPARTMENT OF ENERGY | Office of Science
 Sandia National Laboratories |  **E³SM**
 Energy Exascale Earth System Model

 **SC23**
 Denver, CO | am hpc

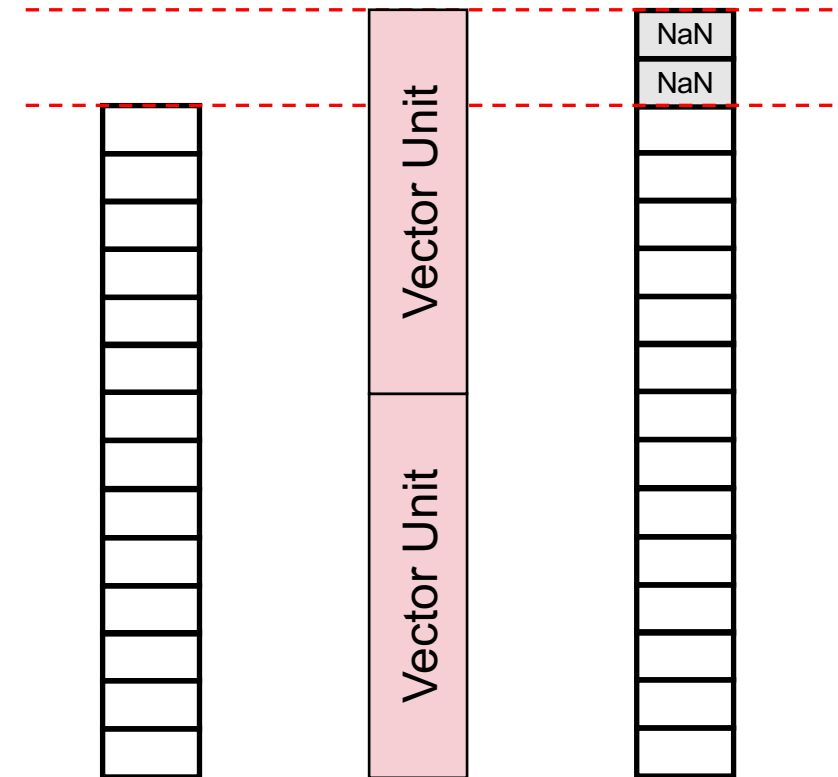
The Simple Cloud-Resolving E3SM Atmosphere Model Running on the Frontier Exascale System

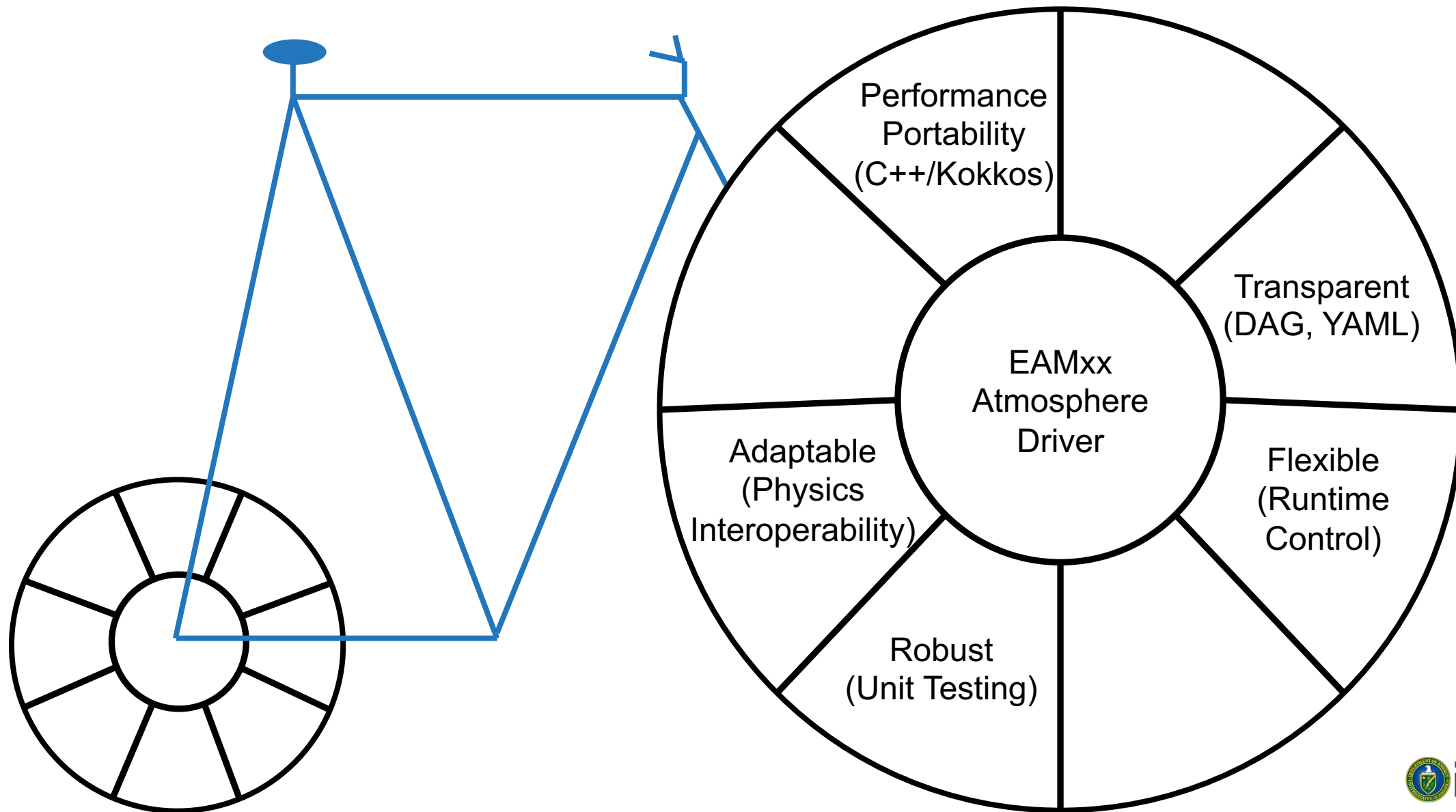
Luca Bertagna (bertag@sandia.gov)
 Sandia National Laboratories
 Nov. 15th 2023

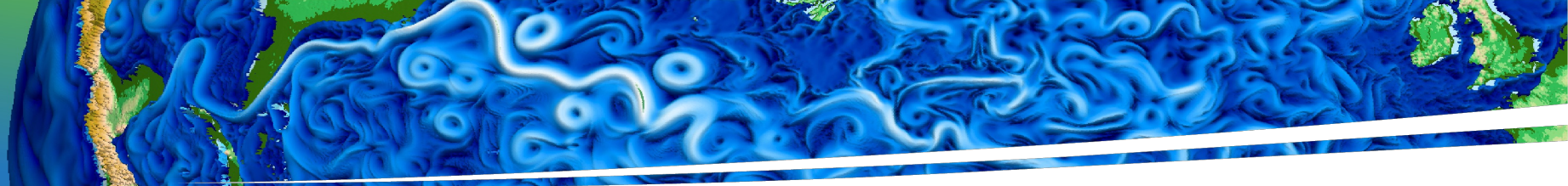
Mark Taylor, Peter Caldwell, Luca Bertagna, Conrad Clevenger, Aaron Donahue, James Foucar, Oksana Guba, Benjamin Hillman, Noel Keen, Jayesh Krishna, Matthew Norman, Sarat Sreepathi, Christopher Terai, James White, Danqing Wu, Andrew Salinger, Renata McCoy, L. Ruby Leung, David Bader

Sandia National Laboratories is a multitechnology laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525

<https://acme-climate.atlassian.net/wiki/spaces/ECM/pages/4129325057/2024-02-15+All-Hands+Presentation+Meeting+Notes>







The new EAMxx paradigm

phys_state
phys_tend
dyn_state
PBUF
cam_in/out



Field Manager Class and Field Objects

- **All persistent variables** are instances of a field object
- All fields managed by a **single Field Manager**
- Simplifies operations on fields, e.g.
 - Remapping to different grids
 - Adding new fields
 - Storing field metadata

cam_init
phys_init
dyn/stepon_init
cam_run,1,2,3,4
phys_run1,2
stepon_run1,2,3



Atmosphere Driver Class and Atmosphere Process Objects

- Every process (**physics and dynamics**) is an instance of an AP object.
- **Atmosphere Driver manages all AP's**, including call sequence order, subcycling, grids, IO
- Able to handle **many runtime configurations**.

The EAM paradigm

- Opaque
- Hardcoded process order,
- Hardcoded time-splitting,
- Difficult to add new processes,
- **Difficult to audit**
- Individual pieces are dependent on the rest of the model.

