CSEG Update:

Introducing CIME

(Common Infrastructure for Modeling the Earth)

Mariana Vertenstein
mvertens@ucar.edu
CESM Software Engineering Group
National Center For Atmospheric Research

NCAR is sponsored by the National Science Foundation

Acknowledgements

- CSEG/CGD
  - Ben Andre
  - Alice Bertini
  - Cheryl Craig
  - Tony Craig
  - Brian Eaton
  - Jim Edwards
  - Erik Kluzek
  - Mike Levy
  - Sheri Mickelson
  - Bill Sacks
  - Sean Santos

- ASAP/CISL
  - Allison Baker
  - John Dennis
  - Sheri Mickelson
  - Kevin Paul

- Others
  - Dorit Hammerling (CISL)
  - Gary Strand (CGD)

Outline

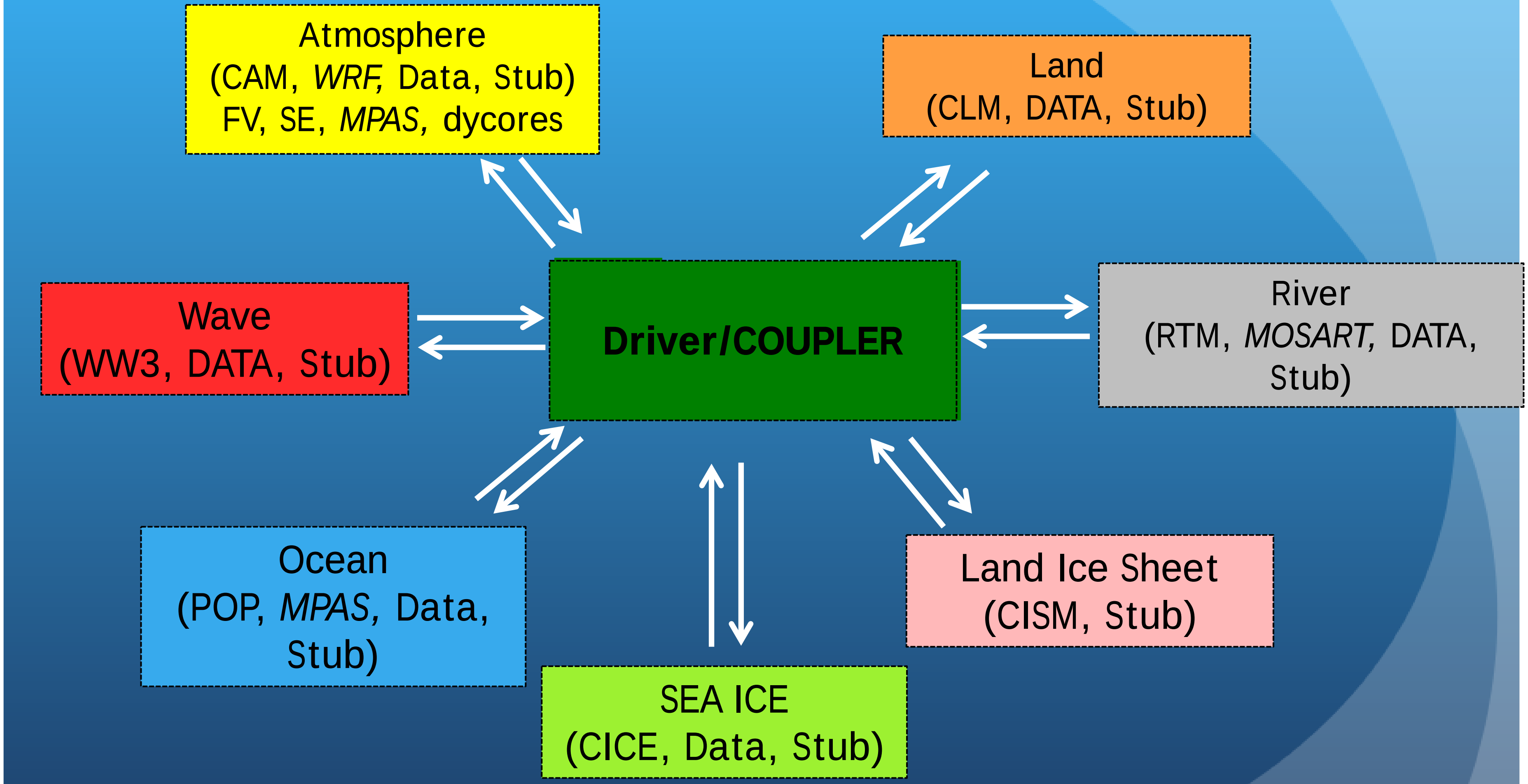
- What is the CESM Infrastructure?
- What is CIME?
- What is the current CIME status?
- What are upcoming CIME developments?

What is CESM Infrastructure?

CESM infrastructure

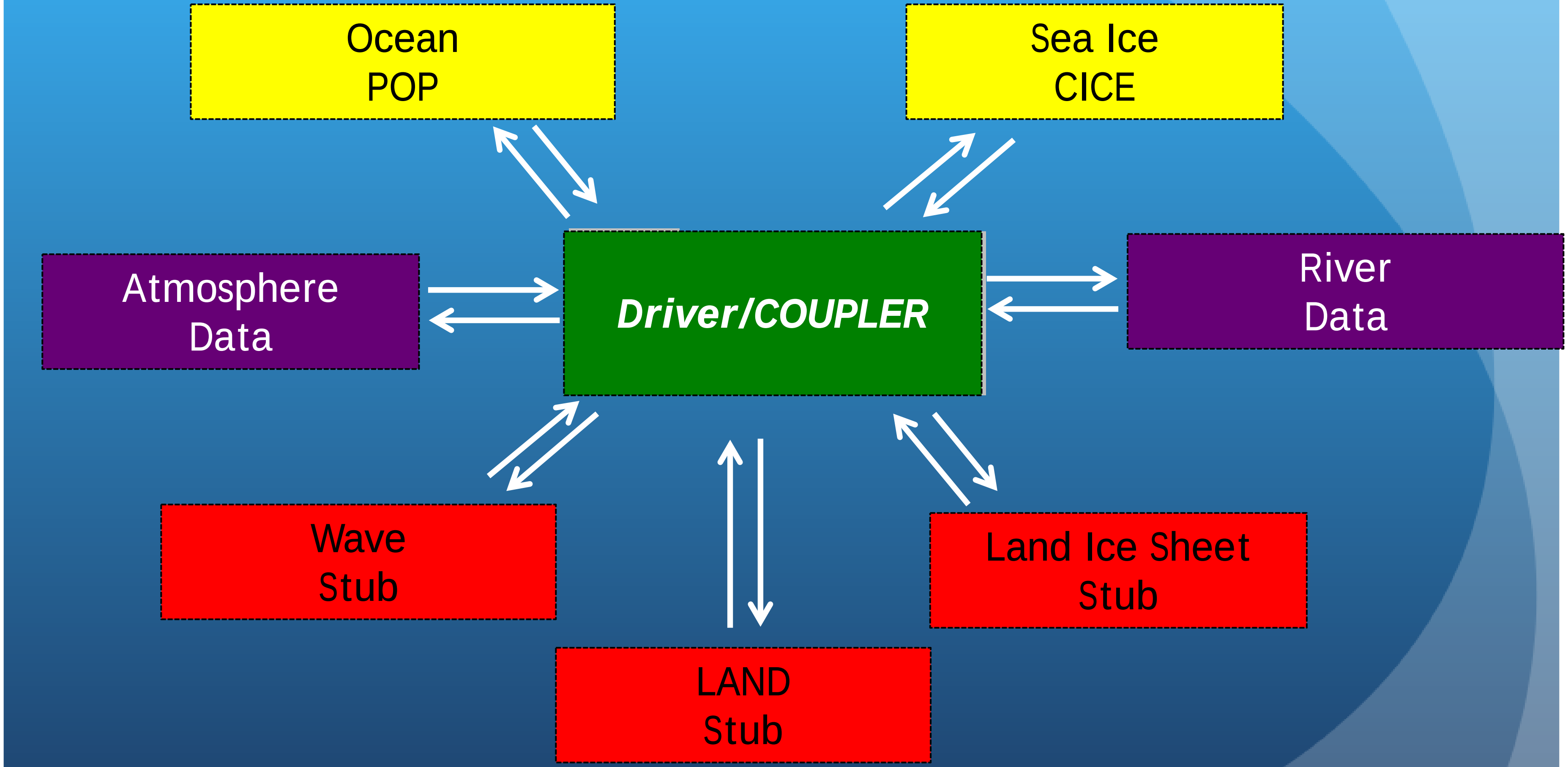
Permits flexible swtiching of

***inter- component** feedbacks – on and off*

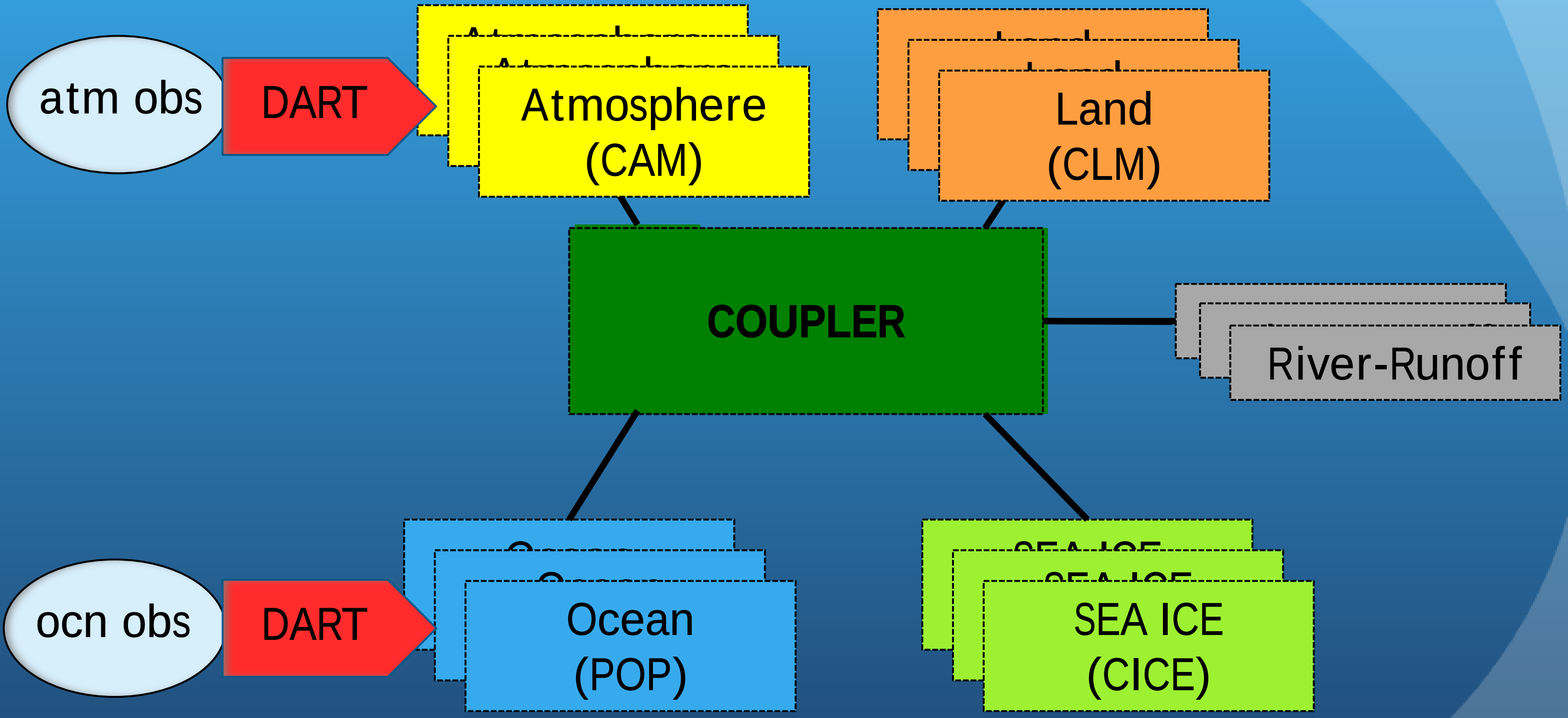


In past infrastructure (no IP) tied to science development (has IP)

CESM infrastructure
Example of core ocean-ice experiments



CIME coupler infrastructure *has multiple instance capability – e.g. used for Ensemble Kalman Filter data assimilation (DART)*



Coupling with DART is presently file based - CESM stops and DART Modifies the atmosphere initial files and ocean restart files

(1) Driver/Coupler Infrastructure

- Targeted to provide a *flexible way to turn off feedbacks* as part of model development and turn on feedbacks for fully coupled simulations – all with ONE coupling infrastructure

- *Agnostic* as to whether a coupling component is prognostic, data or stub

- Components communicate with each other only via a *hub and spoke architecture*

(2) Scripting Infrastructure

- Provides “out-of-the box” case configuration capabilities that allow users to easily turn feedbacks on and off and create experimental configurations with the
  - Target feedbacks enabled
  - Target component grids
  - Target component configurations
- Keeps complexity in the scripts – not dependent on the users having expert knowledge of the entire system
- Provides experimental provenance

(3) Other Infrastructure Utilities

- “PIO (parallel IO)
  - used by all components
- Testing framework
  - Unit and system tests
  - Each can be run with one command
- Share code (e.g. data stream capabilities)
- Statistical Ensemble Test
- Mapping utilities
  - Leverages offline ESMF to create grid mapping files in parallel
- Timing utilities
- MCT (Model Coupling Toolkit)

What is CIME?

Common Infrastructure for Modeling the Earth

Why CIME?

- **Facilitate future infrastructure modernization as a collaborative project (e.g. CESM infrastructure)**

- **Response to February summit of US Global Change Research Program (USGCRP) / Interagency Group on Integrative Modeling (IGIM) as a positive outcome from the February Summit**
  - **IGIM is charged with coordinating global change-related modeling activities across the Federal Government and providing guidance to USGCRP on modeling priorities**

- **Enable separation of infrastructure *(no scientific intellectual property)* versus scientific development codes *(intellectual property that must be protected)***

- **Eliminate duplication of efforts**

Why CIME (cont)

- How can this be accomplished?
  1. Make ALL CESM infrastructure public
  2. Make ALL infrastructure development open
  3. Enable it to have stand-alone capability - developed and tested end-to-end as a stand-alone system (independent of prognostic components)

- This will facilitate AND encourage
  - outside collaboration
  - frequent feedback on infrastructure development
  - quick problem resolution
  - rapid improvement in the productivity, reliability and extensibility of the CIME infrastructure

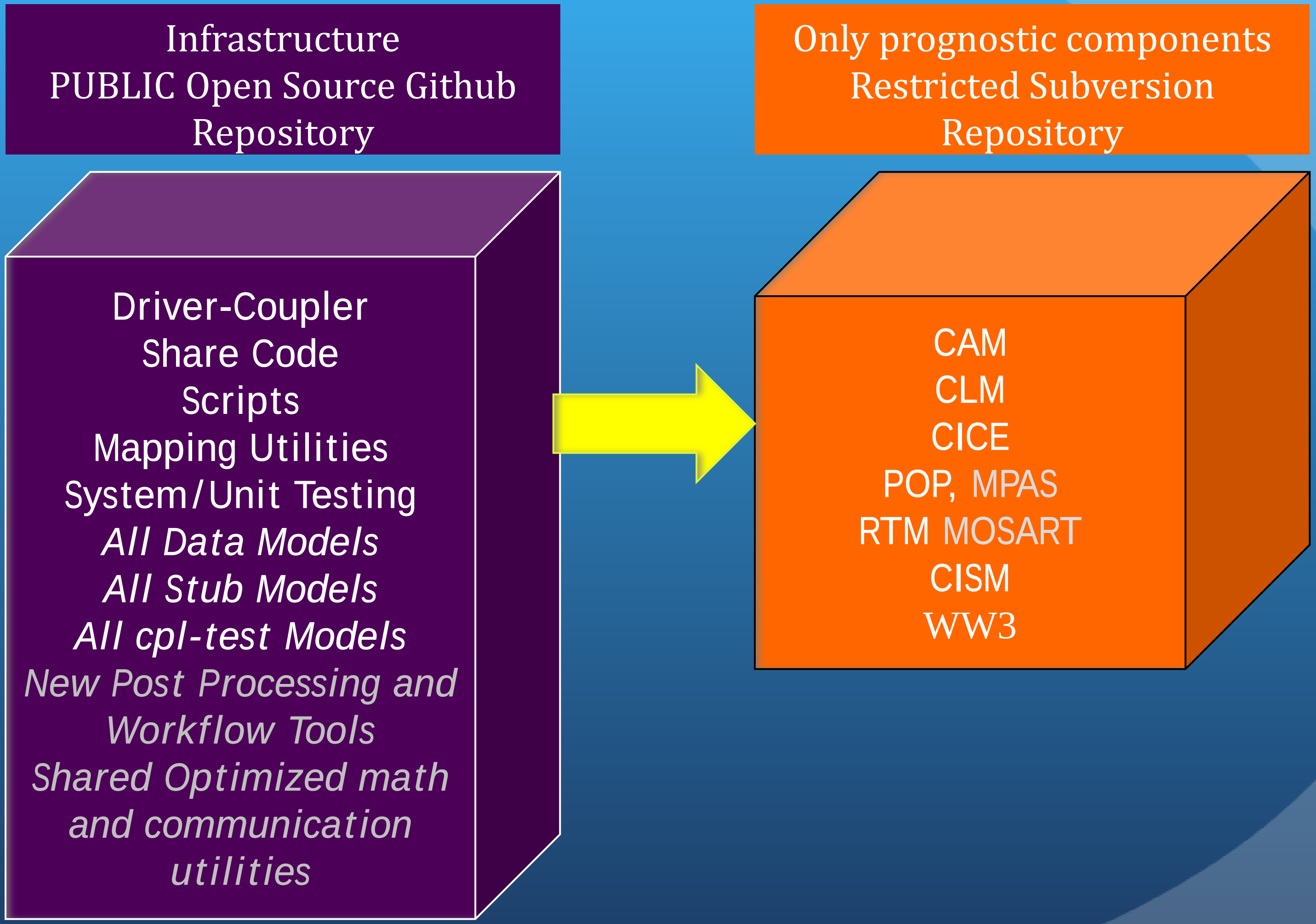
Old paradigm –
everything in ***restricted*** developer repository

Infrastructure and all model components
Restricted Subversion Repository -

Prognostic Models **(CAM, CLM, CICE, etc.)**
Data Models **(DATM, DLND, DICE, etc.)**
Stub Models **(SATM, SLND, SICE, etc.)**
CPL Test Models **(XATM, XLND, XICE, etc.)**

Driver-Coupler Code
Share Code
Scripts
System and Unit Testing
Mapping Utilities
New Post-Processing and Workflow Tools

New paradigm – all infrastructure is ***Open Source (public read) -*** IP still in place for prognostic components



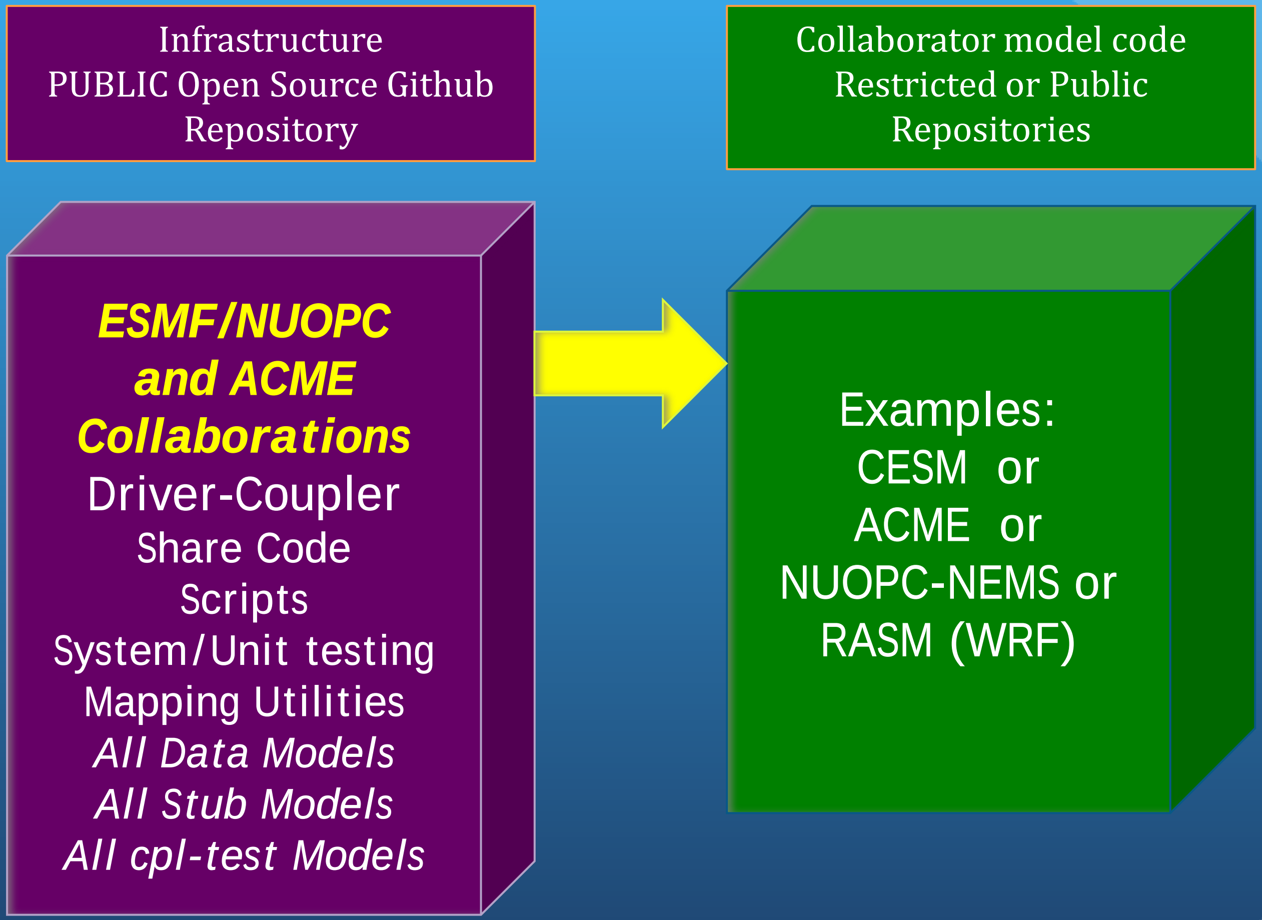
Infrastructure
PUBLIC Open Source Github
Repository

Driver-Coupler
Share Code
Scripts
Mapping Utilities
System/Unit Testing
All Data Models
All Stub Models
All cpl-test Models
New Post Processing and
Workflow Tools
Shared Optimized math
and communication
utilities

Only prognostic components
Restricted Subversion
Repository

CAM
CLM
CICE
POP, MPAS
RTM MOSART
CISM
WW3

CIME Infrastructure can be used to facilitate releases and external collaborations



Infrastructure
PUBLIC Open Source Github
Repository

ESMF/NUOPC
and ACME
Collaborations
Driver-Coupler
Share Code
Scripts
System/Unit testing
Mapping Utilities
All Data Models
All Stub Models
All cpl-test Models

Collaborator model code
Restricted or Public
Repositories

Examples:
CESM or
ACME or
NUOPC-NEMS or
RASM (WRF)

What is CIME Status?

CIME Current Status

- **Stand-alone capability**
  - CIME can be developed, run and tested "stand-alone" with either all data models or all stub-models or all "test-cpl" models
  - End-to-end stand-alone development capability

- ***ALL* CIME development is in the open**
  - Updates to CIME are not are not just snapshots of a private repository. CIME developments are fully visible and anyone can grab them at any time.

- **Consolidation of separate externals**
  - Each part of CIME was previously developed independently and often led to inconsistencies. Now CIME is a ***SINGLE*** entity and ensures consistency among its various parts. This simplifies and adds robustness to infrastructure development process

CIME Status (cont)

- **New unit testing framework**
  - New unit tests in coupler
  - Framework can also be applied to prognostic components
  - CMake based – one line invocation
- **Refactored automated system tests**
  - CIME stand-alone tests accompany CIME
  - Tests to turn off feedbacks (i.e. stand-alone CAM, stand-alone CLM, stand-alone POP are now contained in prognostic component directory) – separation of concerns
- **New workflow capabilities**
  - Introduction of workflow dependencies
  - Short term archiver (movement of model data to local disk for later post-processing) run as a separate job
  - More flexible introduction of new machines

Directory Structure for CIME
2 new top level directories

- **CIME/** (coupled to prognostic components or stand-alone)
  - Driver-Coupler/ *(coupler code base)*
  - Components/data_comps *(data model code base)*
  - Scripts/ *(out of the box supported compsets, grids, pes)*
  - Machines/ *(out of the box supported machines, compilers)*
  - Externals/ *(libraries - pio, mct, gptl, …)*
  - *Post processing Diagnostics/*
- **SRCROOT/**
  - CESM: components/cam/, components/clm/……,
  - ACME: components/acme-atm, components/acme-lnd/
  - NUOPC/HYCOM: components/nuopc-hycom

Paradigm for shared development of CIME ACME/CESM collaboration

CESM gatekeeper controlled:

https://github.com/CESM-Development/cime

CESM/ACME/other gatekeeper controlled:

https://github.com/ESMCI/cime

What are upcoming CIME additions?

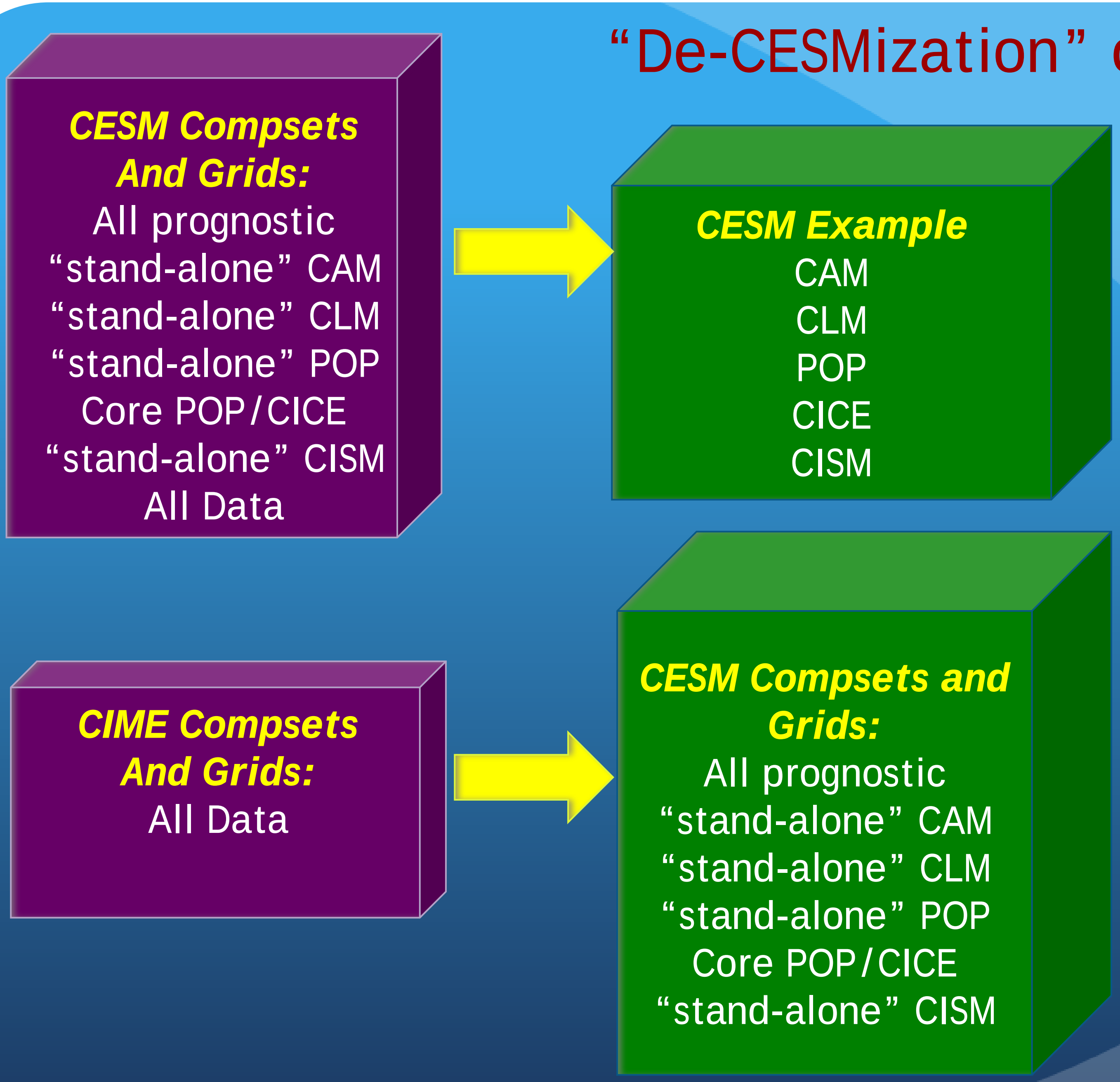
Upcoming additions to CIME

- New parallel IO library - PIO2
- Improved statistical ensemble tests to verify ports and model developments that are non climate changing
  - More in Allison Baker’s SEWG talk
- New end-to-end workflow capabilities (post-processing capabilities as part of a model run)
  - More in Sheri Mickelson’s SEWG talk
- Introduction of new share code libraries
- *“De-CESMization”* of CIME
  - Introduction of requirements and APIs to make new components CIME compatible

Upcoming Shared Code Design
new libraries that can be shared separately

- Separate general shared code library for fortran utilities
  - Provide functionality that fortran does not always provide in a standard way
  - works across a wide range of platforms and compilers
  - Stable APIs
  - Provide performance optimization where appropriate
  - Examples: random number generator, gamma function
- Separate library for scientific share code
  - Examples: saturation vapor pressure, zenith angle calculations
- Libraries would work like PIO, MCT, etc.

“De-CESMization” of CIME



CESM Compsets And Grids:
All prognostic
“stand-alone” CAM
“stand-alone” CLM
“stand-alone” POP
Core POP/CICE
“stand-alone” CISM
All Data

CESM Example
CAM
CLM
POP
CICE
CISM

Current

CIME Compsets And Grids:
All Data

CESM Compsets and Grids:
All prognostic
“stand-alone” CAM
“stand-alone” CLM
“stand-alone” POP
Core POP/CICE
“stand-alone” CISM

Upcoming

Goal is to encourage more community contributions and interactions in the development of CIME

Questions???
Comments???

What do you have to do to become CIME compliant

- A CIME component currently
  - must have init, run, finalize phases
  - only have *one* run phase
- Must have a CIME configuration directory for builds and “component set” definitions
- Must listen to driver for flags for stopping and writing restarts