OASIS4: component model description

Application and component description (XML files):

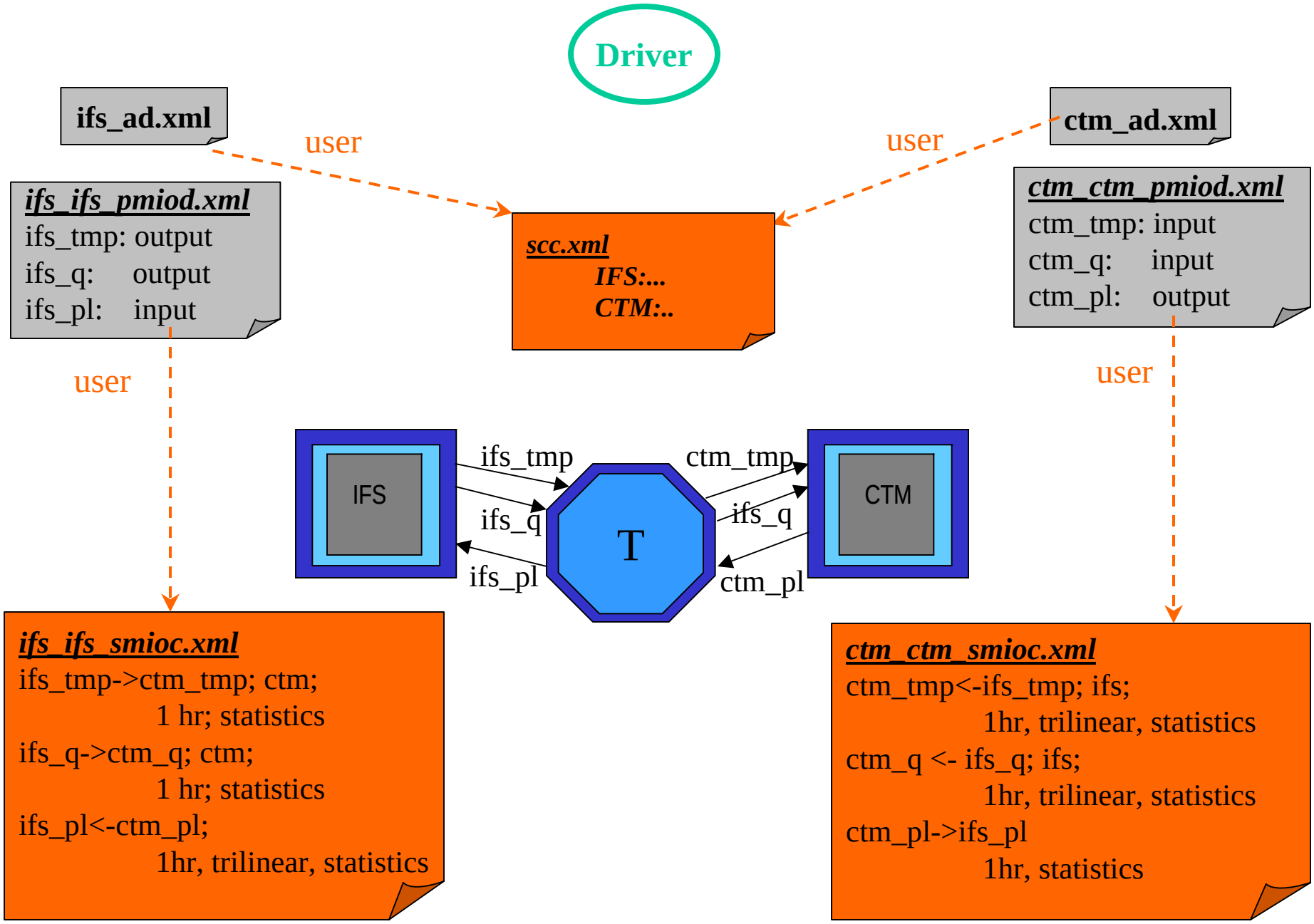
- **For each application** (code):
  - **one Application Description (AD):**
    - possible number of processes, components included, etc.
- **For each component** in the application:
  - **one Potential Model Input and Output Description (PMIOD)**
    - component general characteristics: name, component simulated, …
    - grid information: domain, resolution(s), grid type, …
    - **potential I/O or coupling variables**:
      - local name, standard name (NetCDF CF convention)
      - units, valid min and max
      - numerical type
      - associated grid and points
      - intent –input and/or output

OASIS4: coupled model configuration

Coupled model configuration (XML files):

(Through a GUI,) the user produces

- a **Specific Coupling Configuration (SCC):**
  - start date and end date
  - start mode (MPI1, MPI2)
  - applications, components for each application
  - host(s), number of processes per host, ranks for each component
- **For each component,**
  a Specific Model Input and Output Configuration (SMIOC)
  - **grid** information: **chosen resolution**, …
  - **I/O or coupling variables**:
    - local and standard name, units, valid min max, numerical type, grid
    - **activated intent –input and/or output**
    - **source and/or target** (component and/or file)
    - **coupling or I/O dates**
    - **transformations/interpolations**


Driver
ifs_ad.xml
user
ctm_ad.xml
user
ifs_ifs_pmiod.xml
ifs_tmp: output
ifs_q: output
ifs_pl: input
scc.xml
IFS:...
CTM:..
ctm_ctm_pmiod.xml
ctm_tmp: input
ctm_q: input
ctm_pl: output
user
user
IFS
ifs_tmp
ifs_q
ifs_pl
T
ctm_tmp
ifs_q
ctm_pl
CTM
ifs_ifs_smioc.xml
ifs_tmp->ctm_tmp; ctm;
1 hr; statistics
ifs_q->ctm_q; ctm;
1 hr; statistics
ifs_pl<-ctm_pl;
1hr, trilinear, statistics
ctm_ctm_smioc.xml
ctm_tmp<-ifs_tmp; ifs;
1hr, trilinear, statistics
ctm_q <- ifs_q; ifs;
1hr, trilinear, statistics
ctm_pl->ifs_pl
1hr, statistics

Grid Definition

PRISM_def_grid (**grid_id**, grid_name, comp_id, shape, type, ierr)

• PRISM_reglonlatvrt	lon(i)	lat(j)	height(k)
• PRISM_irrlonlat_regvrt	lon(i,j)	lat(i,j)	height(k)
• PRISM_irrlonlatvrt	lon(i,j,k)	lat(i,j,k)	height(i,j,k)
• PRISM_gaussreduced_regvrt	lon(nhor)	lat(nhor)	height(k)
• PRISM_irrlonlat_sigmavrt	lon(i,j)	lat(i,j)	height(i,j,k)
• PRISM_reglonlat_sigmavrt	lon(i)	lat(j)	height(i,j,k)
• PRISM_unstructlonlat_regvrt	lon(nhor)	lat(nhor)	height(k)
• PRISM_unstructlonlat_sigmavrt	lon(nhor)	lat(nhor)	height(nhor,k)
• PRISM_unstructlonlatvrt	lon(npts)	lat(npts)	height(npts)

Grid Corner Definition

• PRISM_reglonlatvrt	lon(i,2)	lat(j,2)	height(k,2)
• PRISM_irrlonlat_regvrt[1]	lon(i,j,$n_{1/2}$)	lat(i,j, $n_{1/2}$)	height(k,2)
• PRISM_irrlonlatvrt[3]	lon(i,j,k,n)	lat(i,j,k,n)	height(i,j,k,n)
• PRISM_gaussreduced_regvrt[2]	lon(nhor,2)	lat(nhor,2)	height(k,2)
• PRISM_irrlonlat_sigmavrt[2]	lon(i,j,$n_{1/2}$)	lat(i,j,$n_{1/2}$)	height(i,j,k,n)
• PRISM_reglonlat_sigmavrt[2]	lon(i,2)	lat(j,2)	height(i,j,k,n)
• PRISM_unstructlonlat_regvrt[2]	lon(nhor,$n_{1/2}$)	lat(nhor,$n_{1/2}$)	height(k,2)
• PRISM_unstructlonlat_sigmavrt[2]	lon(nhor,$n_{1/2}$)	lat(nhor,$n_{1/2}$)	height(nhor,k,n)
• PRISM_unstructlonlatvrt[3]	lon(npts,n)	lat(npts,n)	height(npts,n)

[1] : corners given with Fortran ordering
[2] : corners given counterclockwise looking in the k positive direction
[3]: convention needed to specify corner order

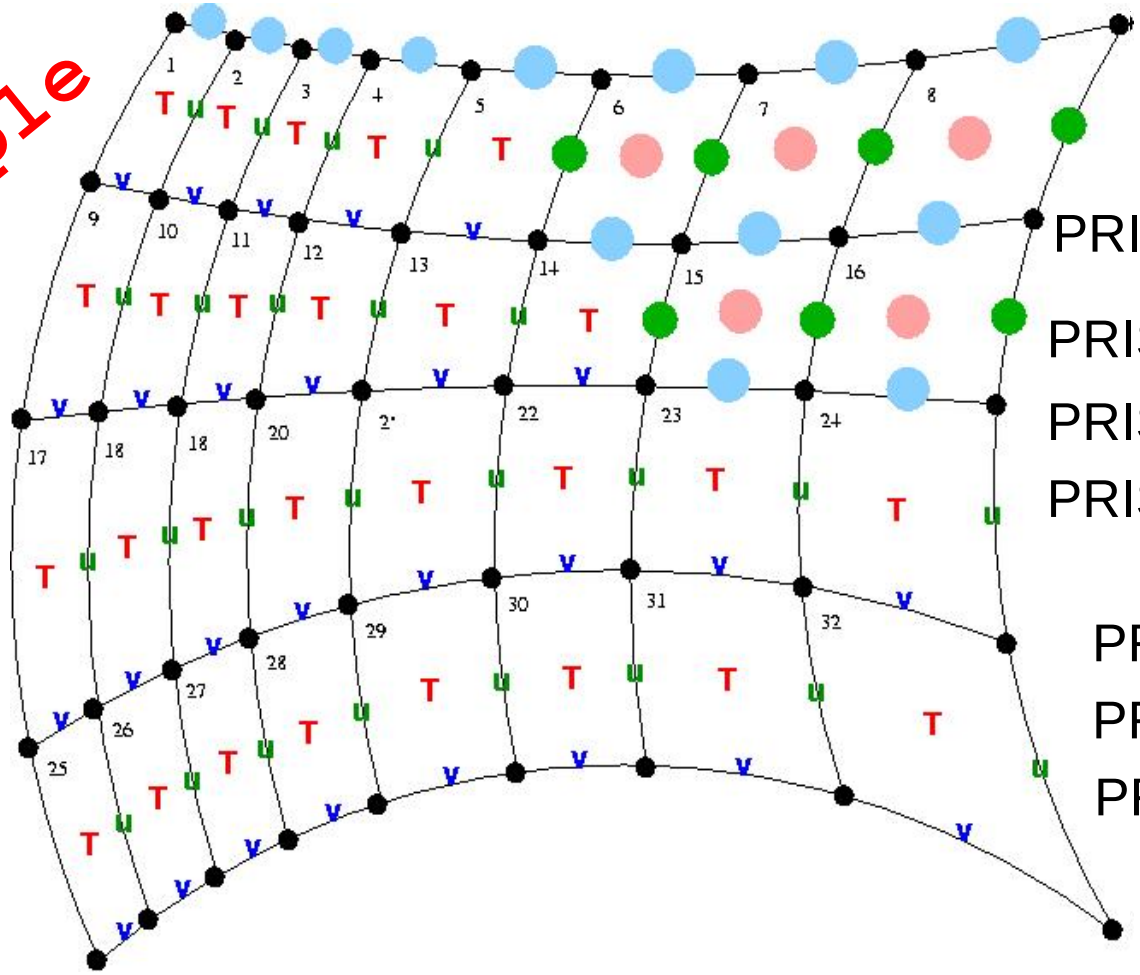
Grid Definition

Other terms in use to describe:

- "regular lat-lon": constant delta in latitude, constant delta in longitude
- "irregular lat-lon": lon(i), lat(j)
- "logically-rectangular": lon(i,j), lat(i,j) (includes reg and irreg lat lon grids)
- "stretched, rotated": lon(i,j), lat(i,j)
- "lat-lon Gaussian grid": constant delta in longitude, lat(j)
- "Gaussian reduced grid": each latitudinal circle split into a different number of cells

PSMILe Grid Definition

Example



PRISM_def_grid

PRISM_set_corners

PRISM_set_points

PRISM_set_points

PRISM_set_points

PRISM_set_mask

PRISM_set_mask

PRISM_set_mask