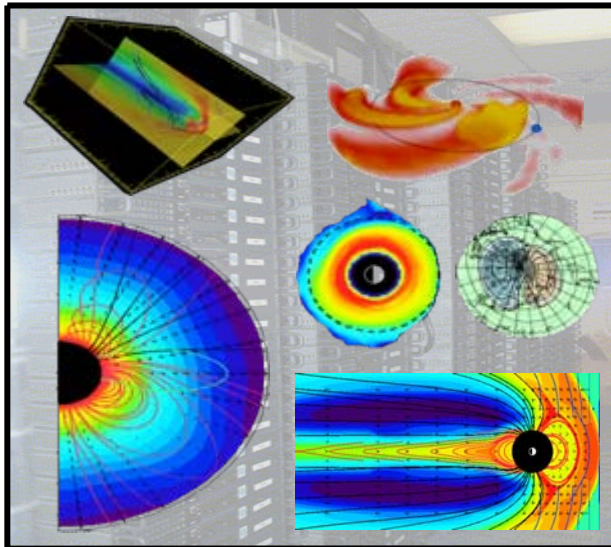


CCMC Data Standards

Conversion, Access, & Interpolation
Using Kameleon

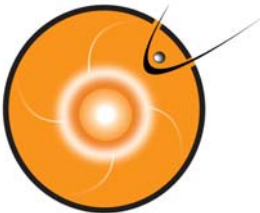


Marlo Maddox

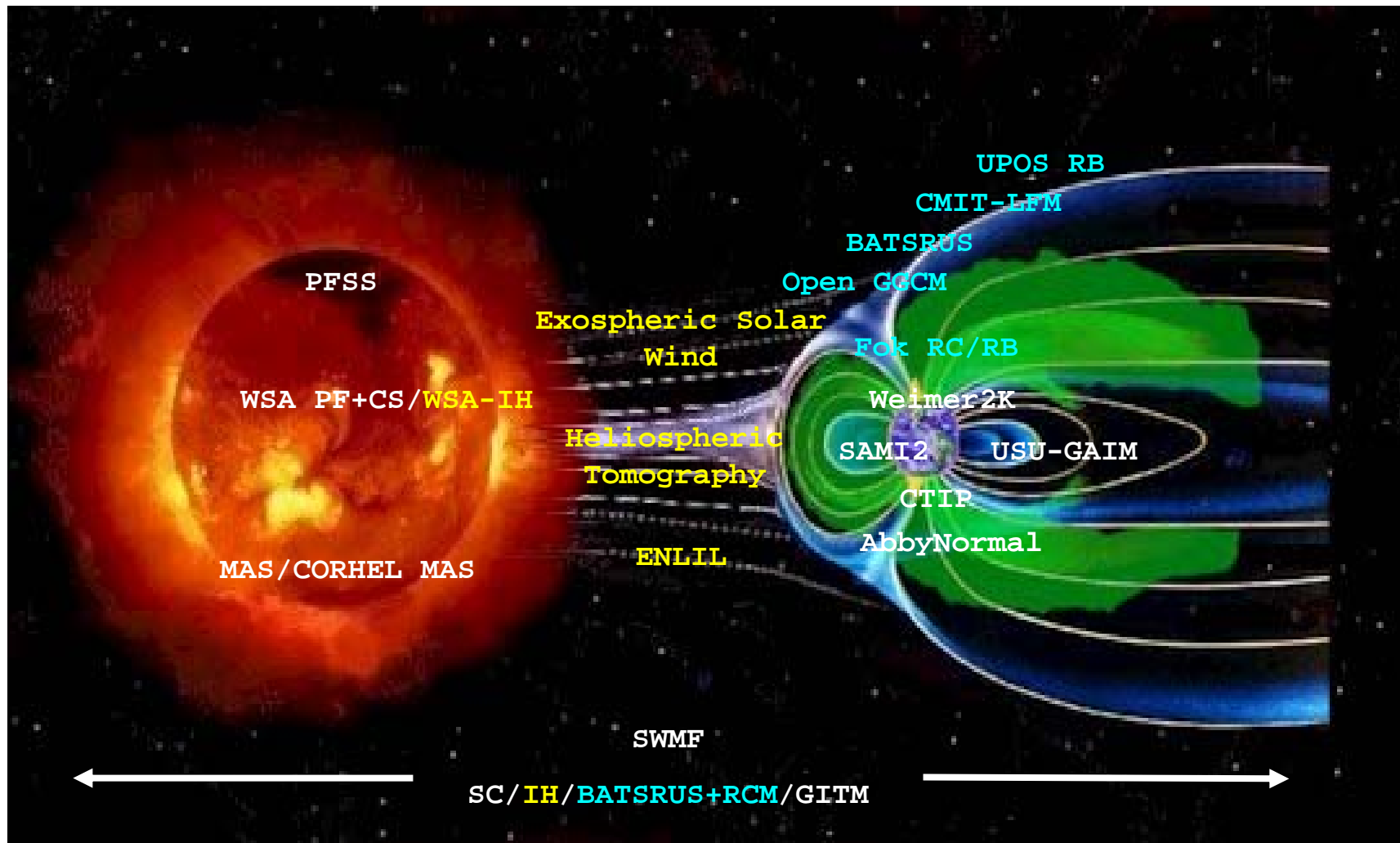
<http://ccmc.gsfc.nasa.gov>

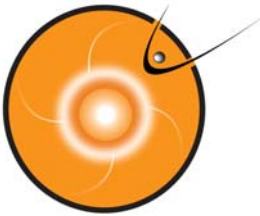
NASA Goddard Space Flight Center





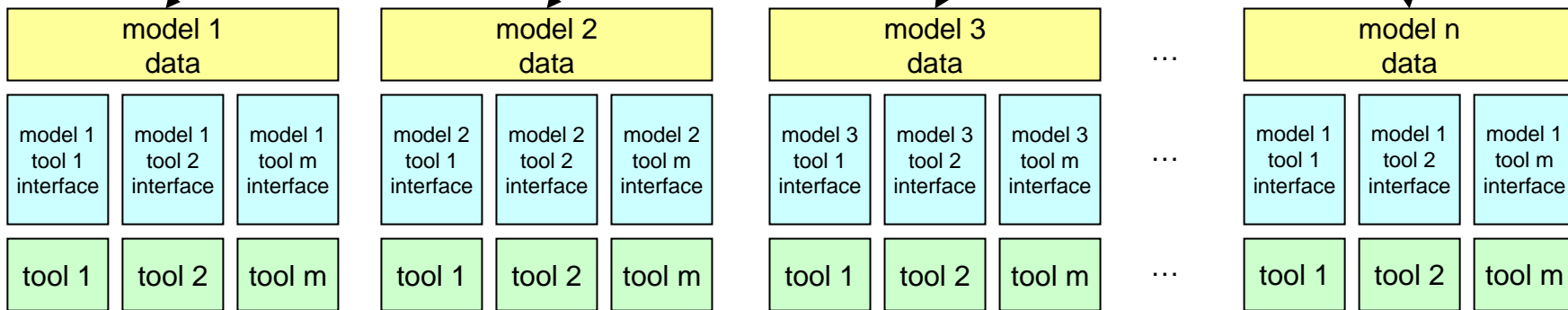
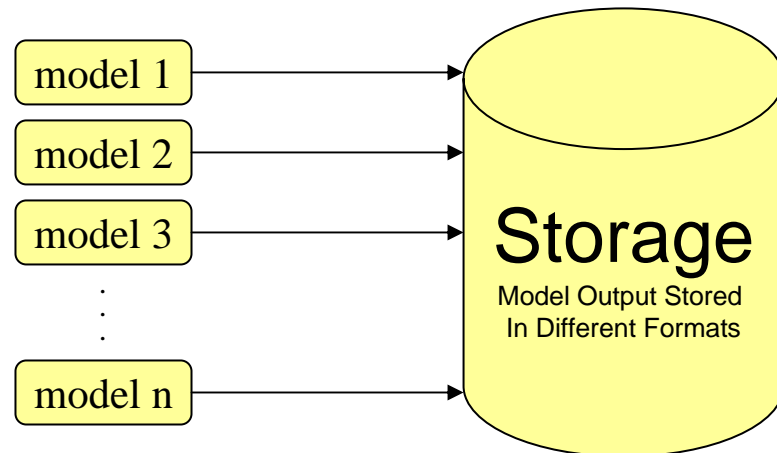
Models Covering the Entire Domain



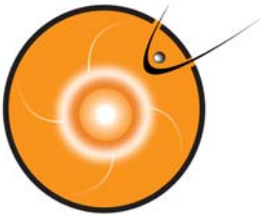


Working With Unique Model Output

- No rules for standard model interfaces
- Each new model has unique output format
- Developer/user needs to become familiar with internal structure of each output file
- Custom read routines to access model data
- Data typically is not self descriptive
- Reduces portability and reuse of
 - Data output itself
 - Tools created to analyze data

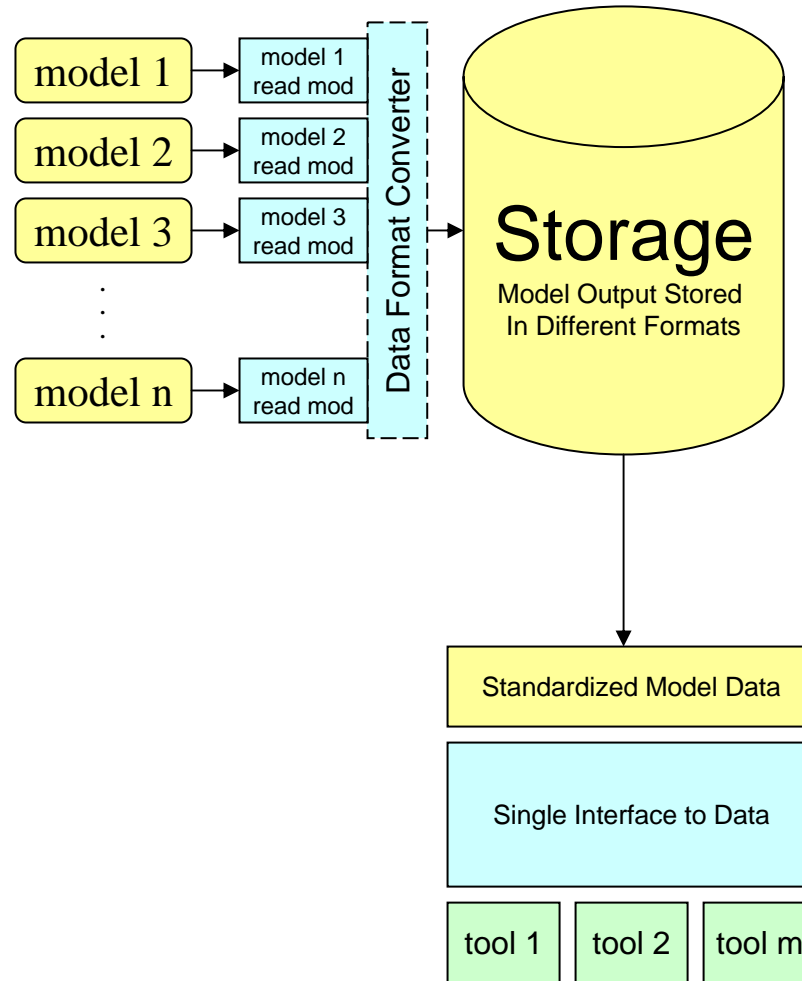


n x m interfaces required

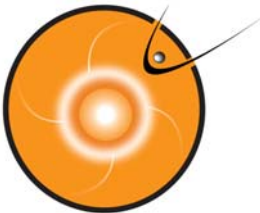


Standardizing Unique Model Output

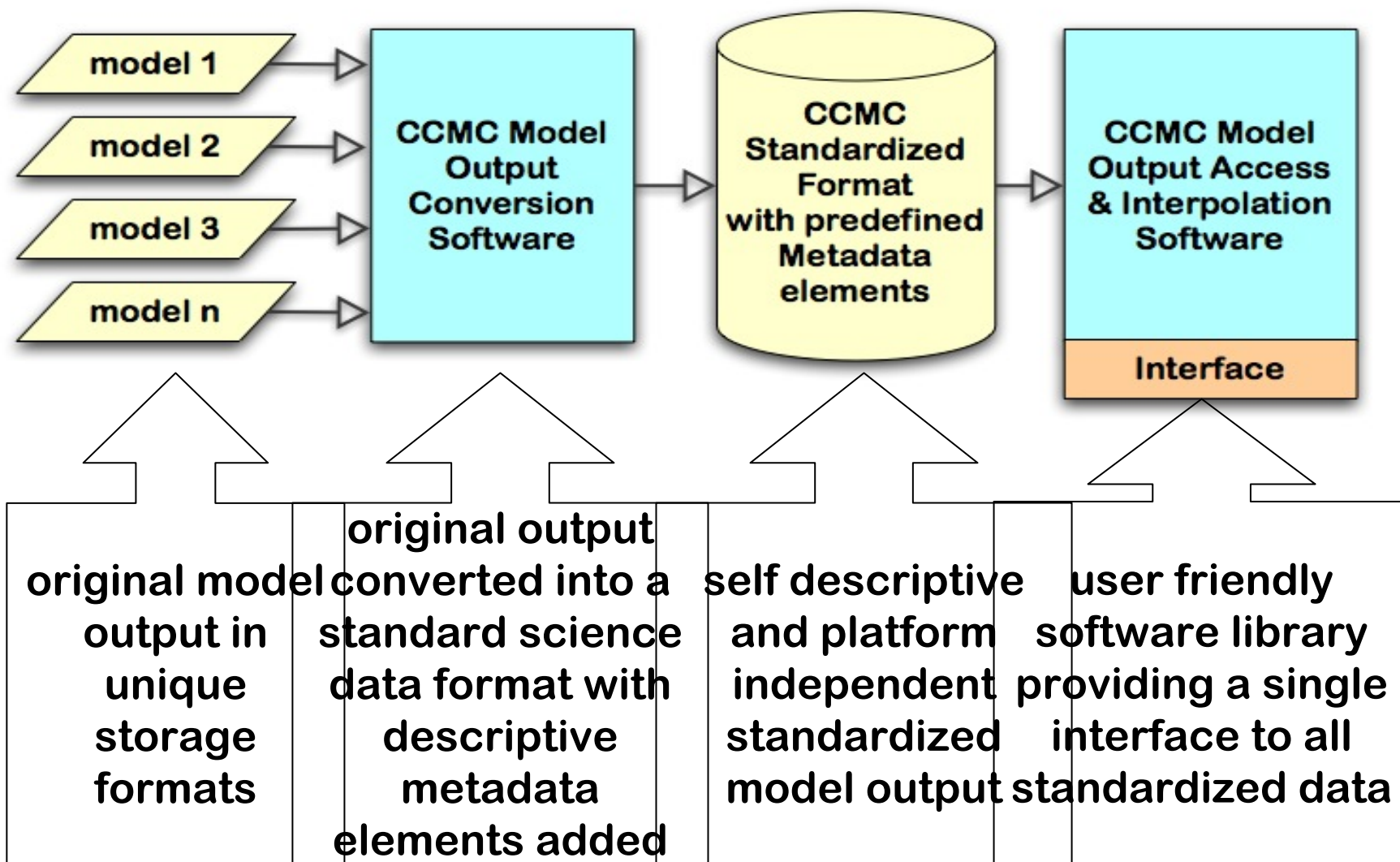
- Original output can be preserved
- Standard format for storage, coupling, visualization, & dissemination
- Model developers continue to have freedom of choice
- Ensures compatibility between models for coupling
- Ground work for which standard, reusable interfaces and tools can be developed

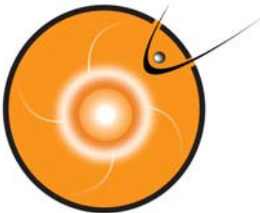


$n + m$ interfaces required



Handling Unique Model Output With Kameleon





Kameleon Supported Models At The CCMC

Standardized Model Output



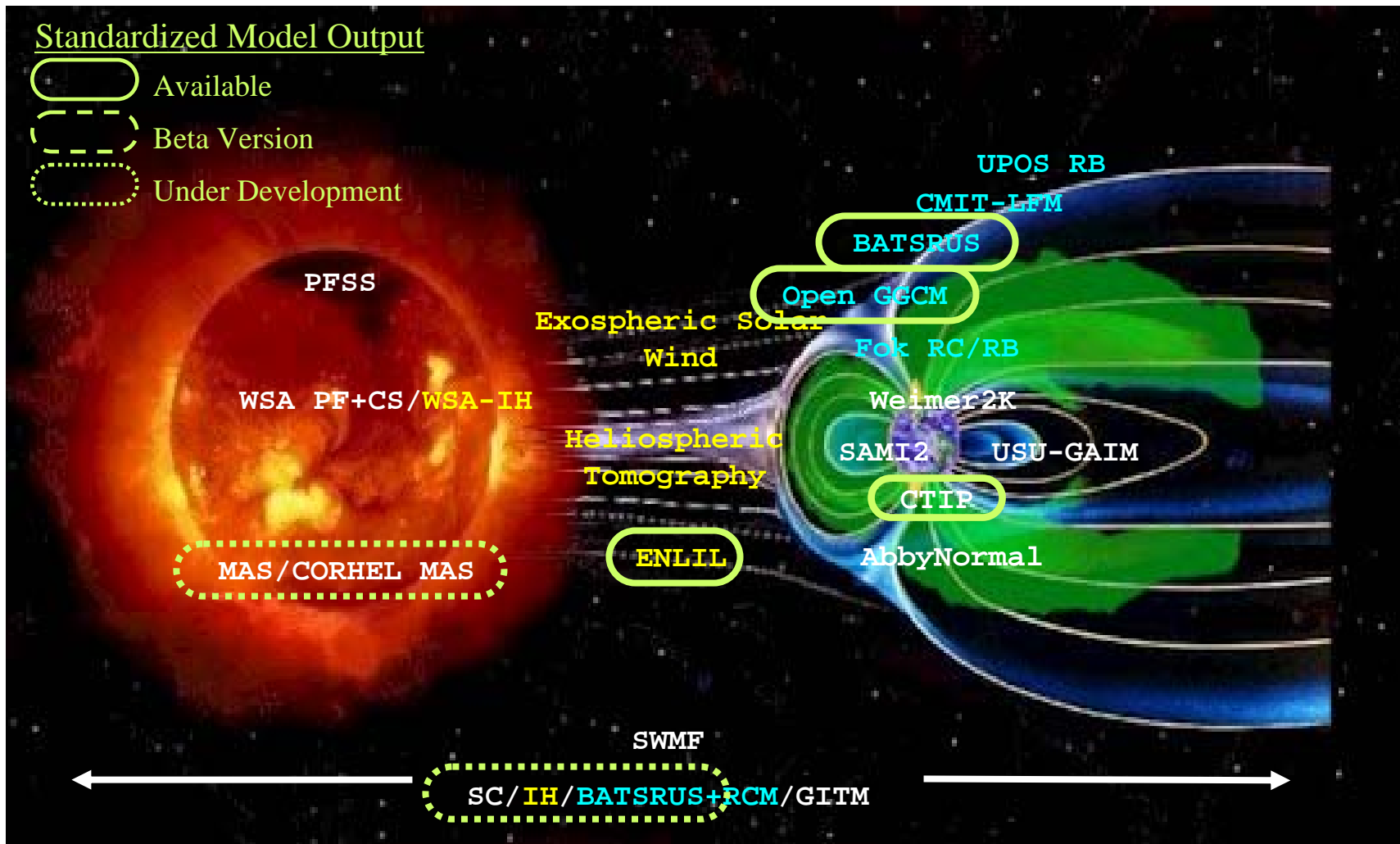
Available

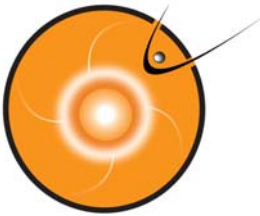


Beta Version



Under Development

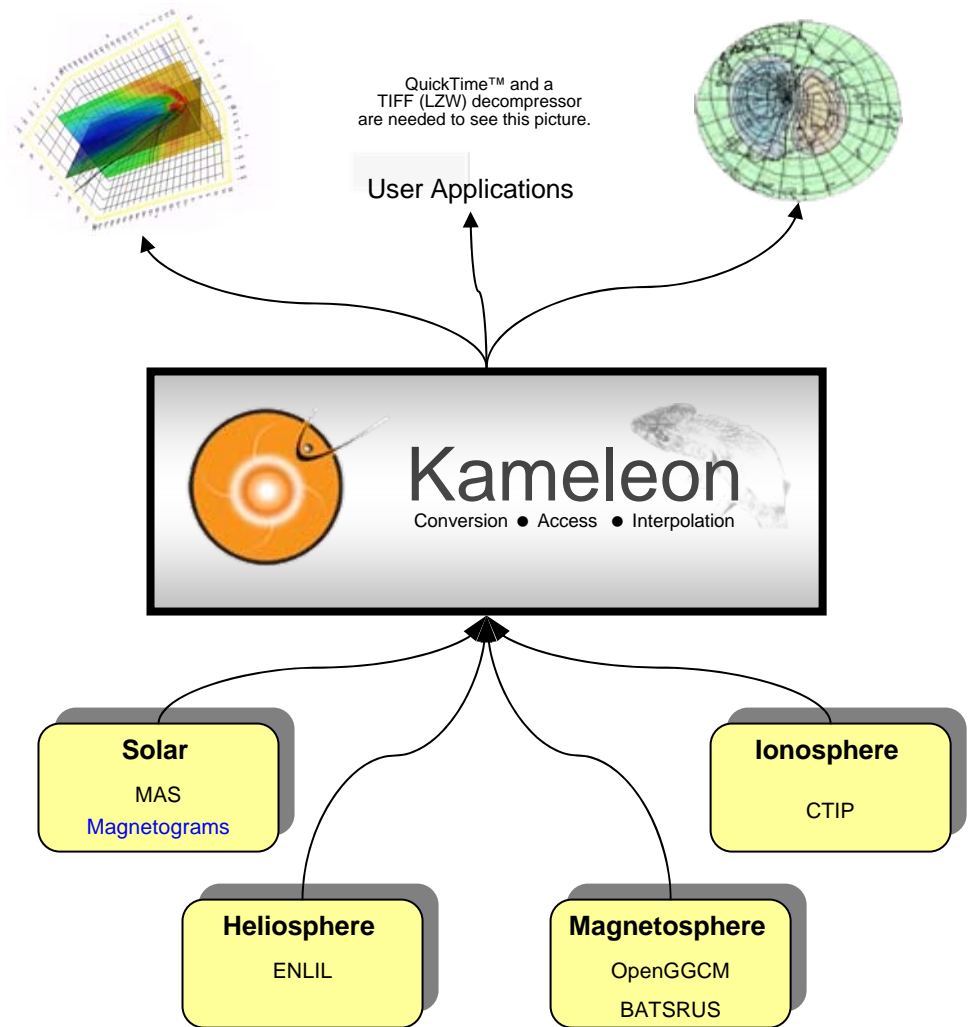




Kameleon Software Suite Overview

Kameleon Software Suite

- Converts and stores disparate data sets into self-descriptive standardized files
- Comprehensive metadata model applied to each file
- Library provides direct data access to converted space weather data
- Interpolation, metadata extraction, & derived variable calculations
- Library callable from any C-supported programming language or application
- Promotes data reuse & code reuse
- Can support/be applied to more than model output - Magnetogram Synthesis (P. Macneice)



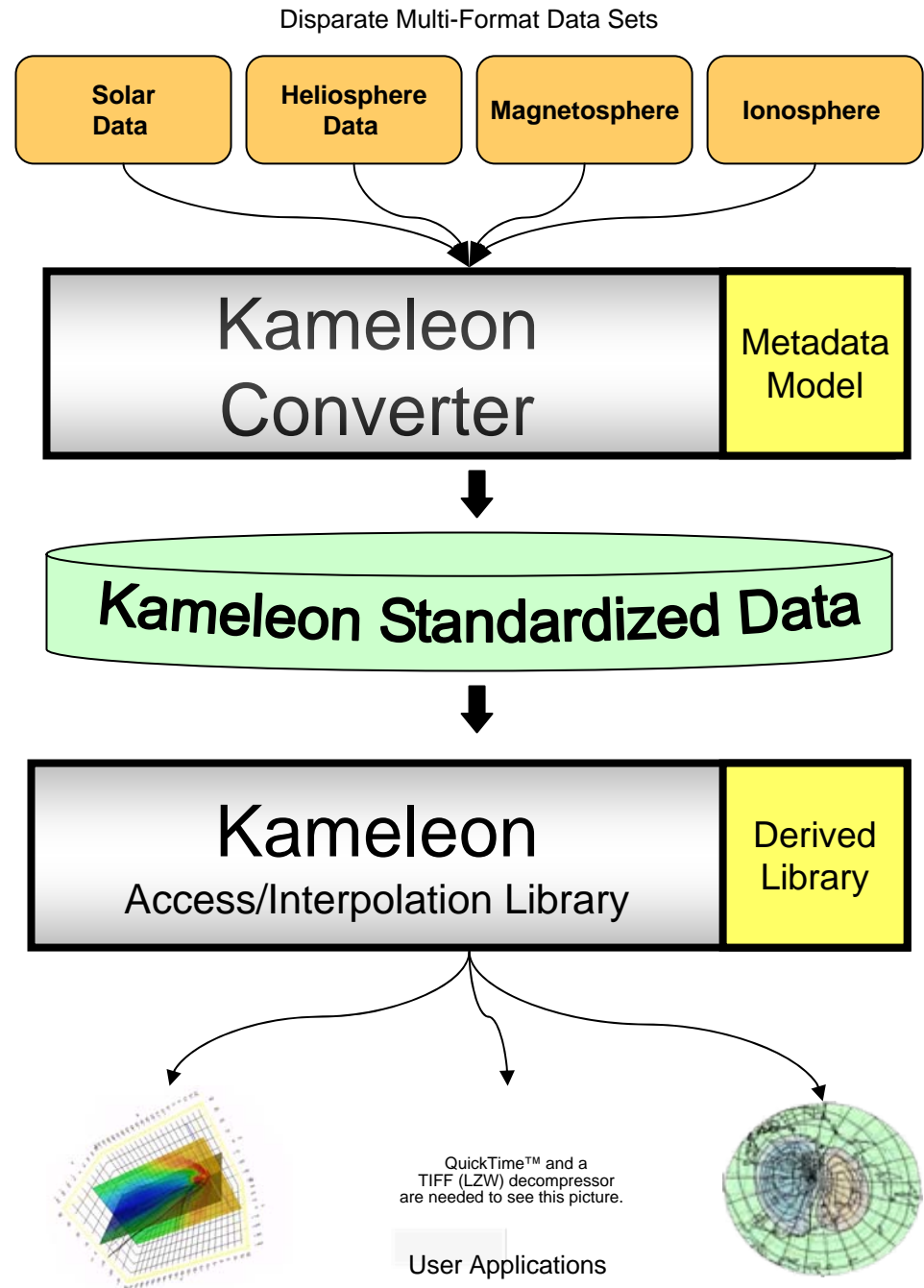
Disparate Multi-Format Data Sets

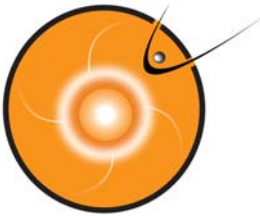
Kameleon Converter

- Ingests supported data files and converts original data into a specific scientific data standard format
 - Platform independent
- Adds descriptive meta elements to each converted data files
 - Grid description Information
 - Coordinate System descriptions
 - Detailed variable descriptors
 - General and Model specific descriptive information

Kameleon Library

- Provides access and interpolation functionality to Kameleon converted data files
 - Standard interface to Multiple and diverse data sets
 - Masks complexity of underlying storage container
 - Efficient direct data access
- Spatial & temporal interpolation
- Query global & variable metadata attributes
- Several interfaces Provided
 - C, C++, FORTRAN, IDL
- Can be used in any C supporting application





Metadata

- Aside from the one-to-one data conversion, what additional metadata do we want to provide?
- Global
 - General description of the model / data
 - Coordinate system(s)
 - Grid Description
 - # of grids
 - # of dimensions
 - dimension size(s)
 - Date & Time Information
- Variable metadata - descriptive elements for each variable
 - Units
 - Actual & Valid Min/Max values
 - Masks Values
- Model Specific Metadata
- SPASE - Space Physics Archive Search and Extract Data Model
 - Computational Model Group
- UMICH - SWMF / Batsrus Data Standardization

Kameleon Global Attributes

- ◉ README
- ◉ README_visualization
- ◉ model_name
- ◉ model_type
- ◉ generation_date
- ◉ original_output_file_name
- ◉ run_registration_number
- ◉ generated_by
- ◉ terms_of_usage
- ◉ grid_system_count
- ◉ grid_system_n_number_of_dimensions
- ◉ grid_system_n_dimension_m_size
- ◉ grid_system_n
- ◉ output_type
- ◉ standard_grid_target
- ◉ grid_n_type
- ◉ start_time
- ◉ end_time
- ◉ run_type

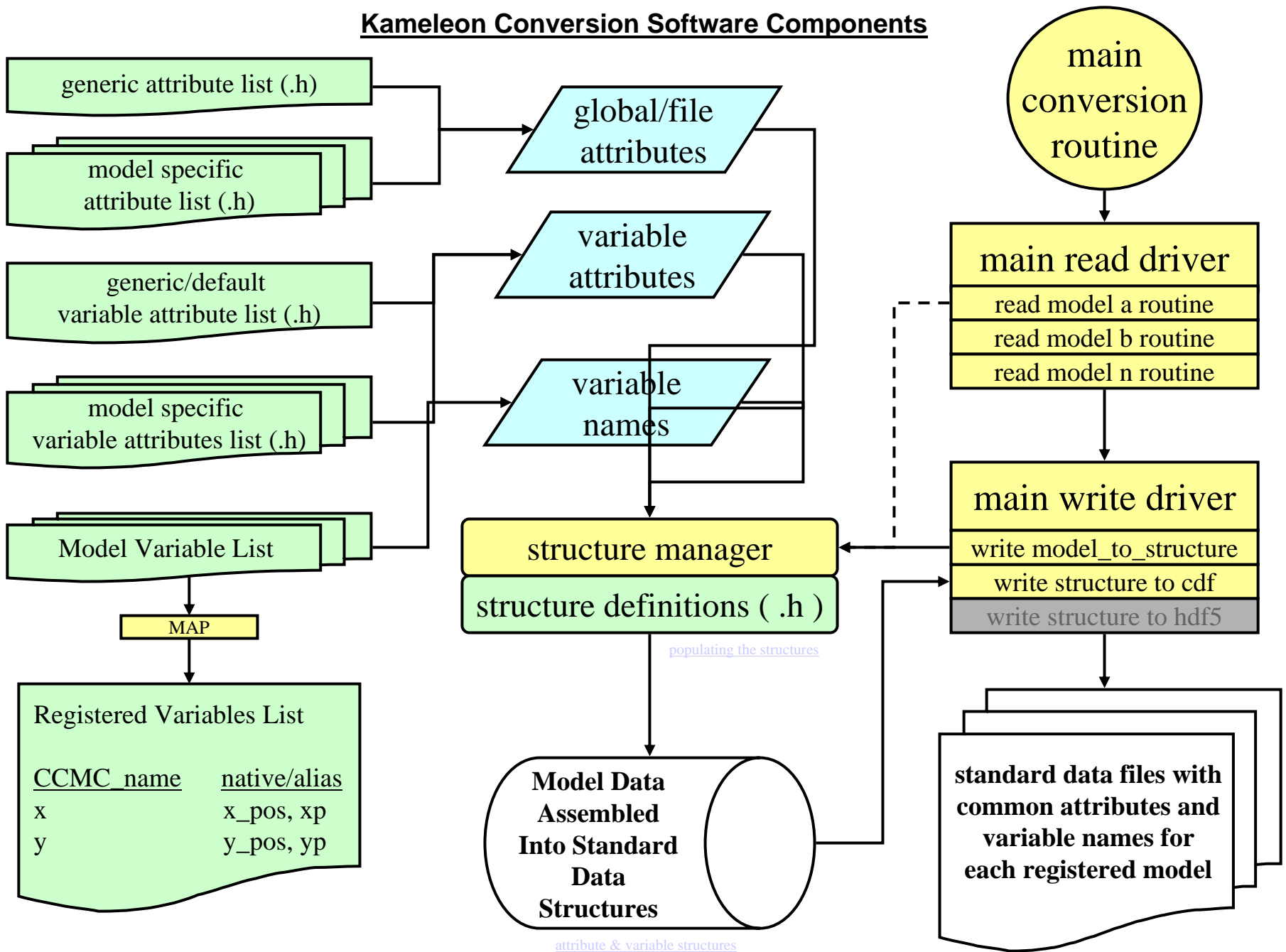
Kameleon Variable Attributes

- ◉ valid_min
- ◉ valid_max
- ◉ units
- ◉ grid_system
- ◉ mask
- ◉ description
- ◉ is_vector_component
- ◉ position_grid_system
- ◉ data_grid_system
- ◉ actual_min
- ◉ actual_max

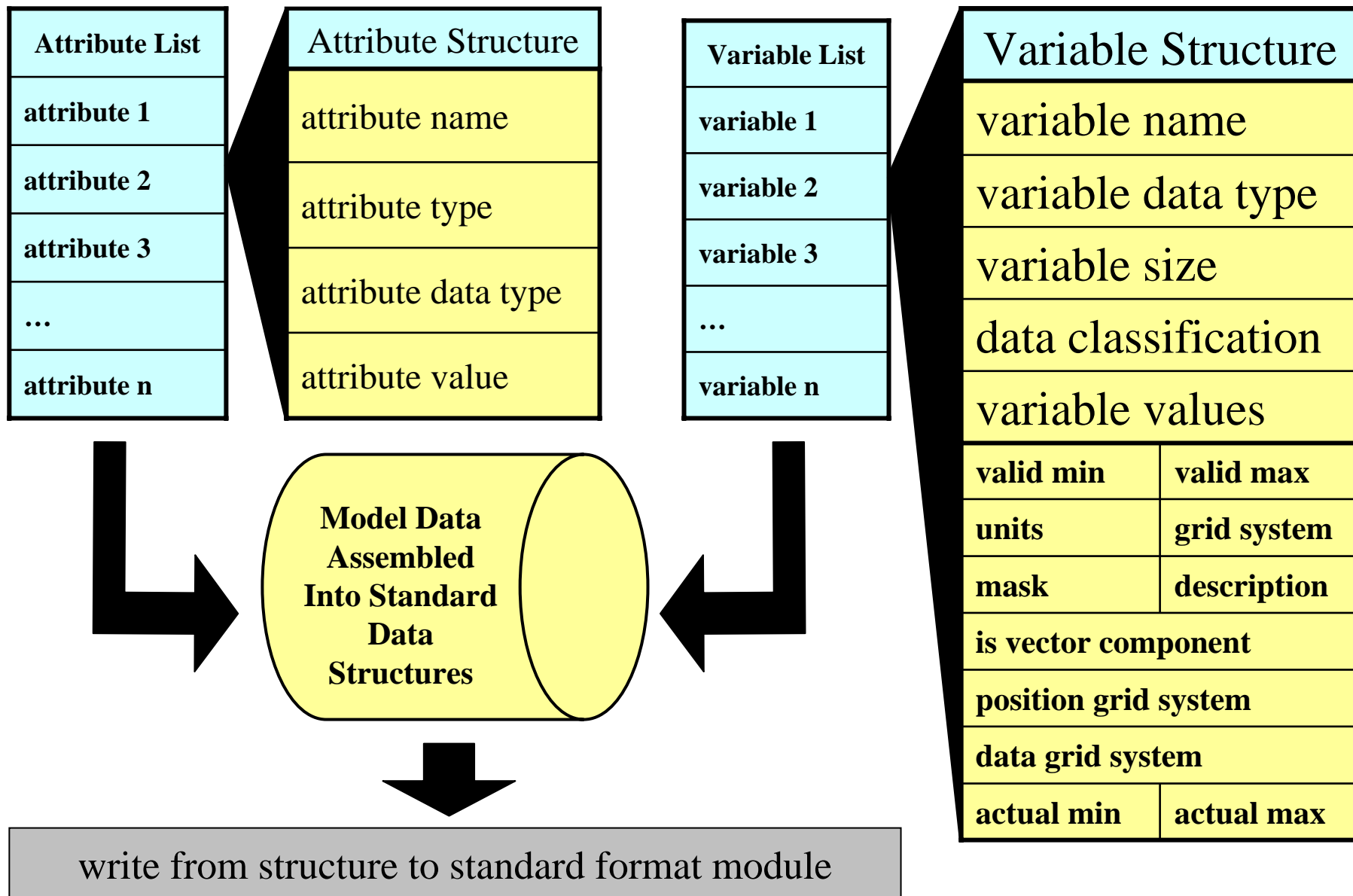
Model Specific Attributes

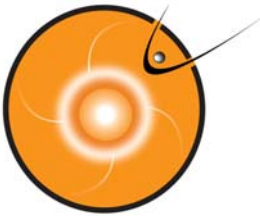
- ◉ Additional grid descriptors
- ◉ Original output data or descriptors that don't map to predefined attributes
- ◉ Any additional elements that are specific or unique to a particular model or space weather domain

Kameleon Conversion Software Components



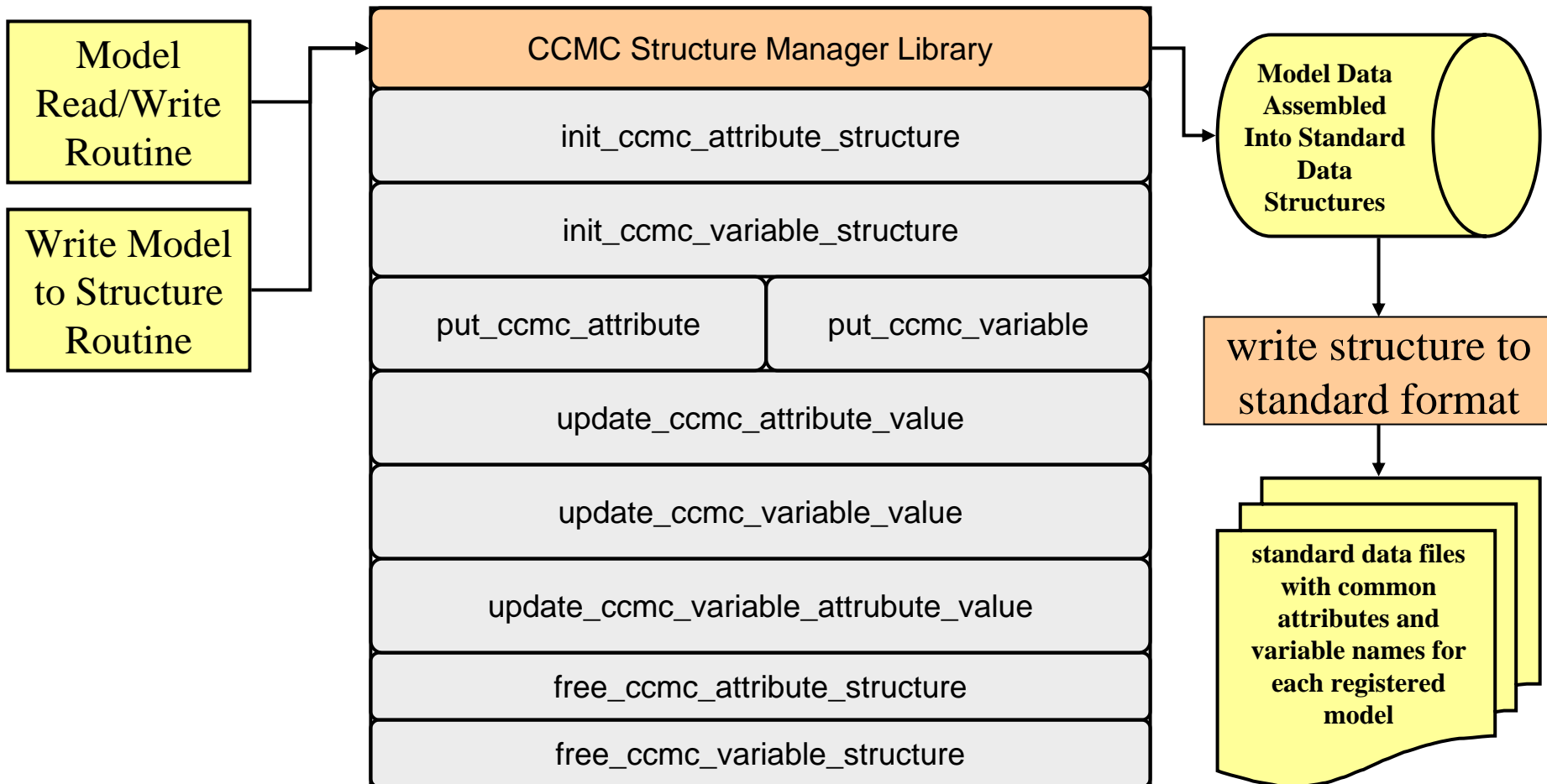
Standardized Attribute & Variable Structure Lists



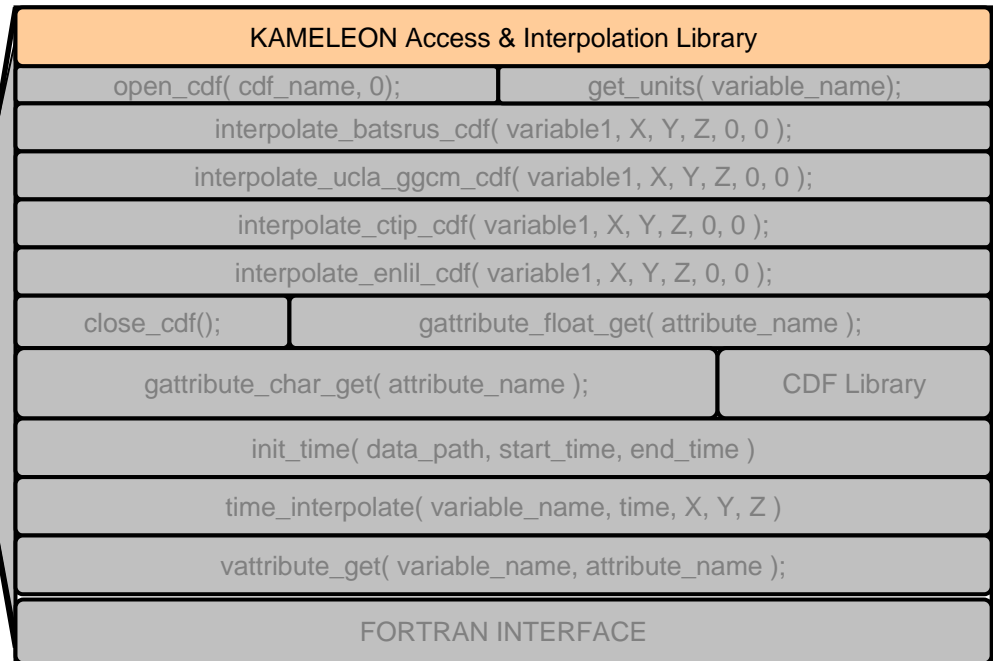
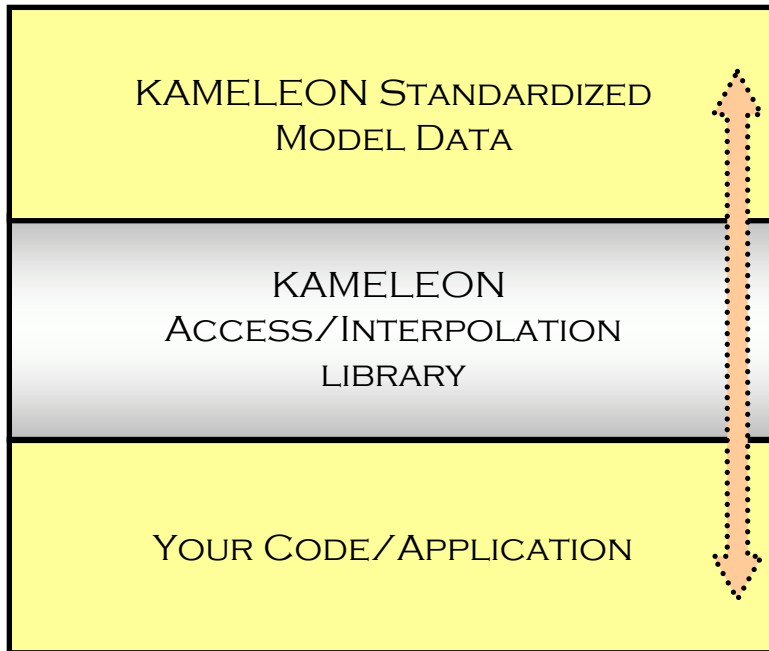


Populating the Structures

- Library of C routines that are used to populate the standard attribute and variable structures.



KAMELEON Access/Interpolation Library



Call from any C supported Programming Language:

- Fortran
- C/C++
- IDL
- OpenDx
- Java
- Perl
- Vtk
- Your App

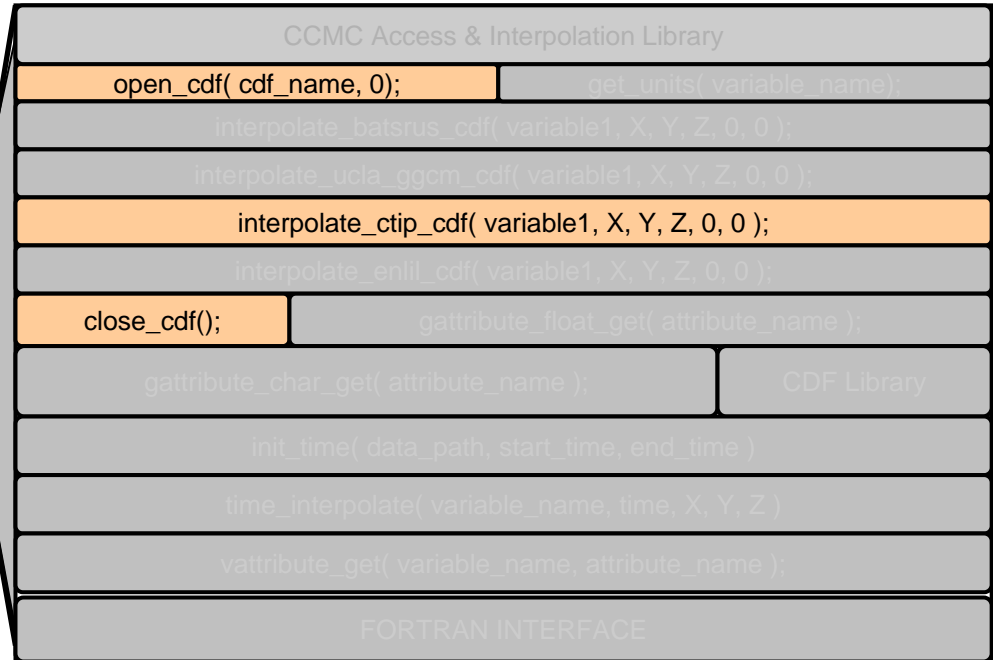
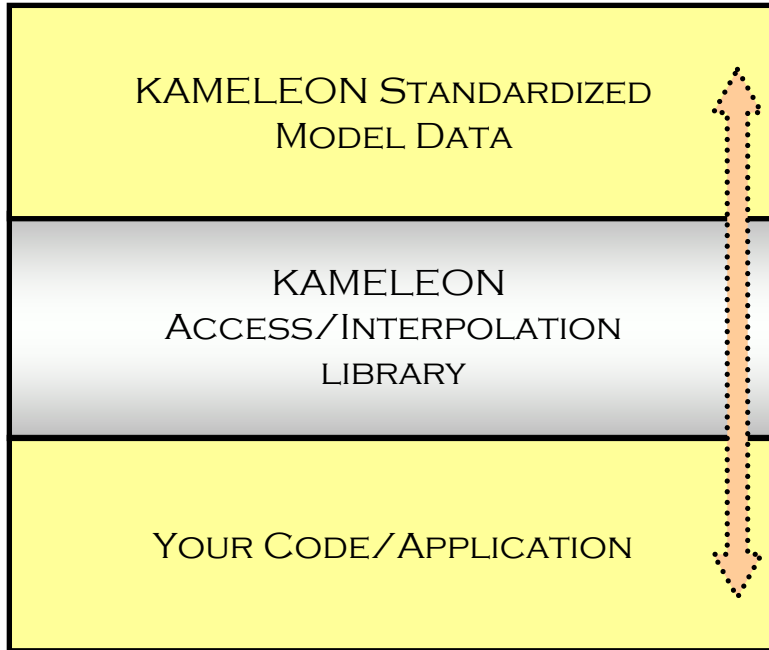
Current Standardized Model Output Availability

- BATSRUS
- OpenGGCM / UCLA-GGCM
- CTIP
- ENLIL
- MAS (*Beta Version*)

Currently Supported Science Data Formats

- CDF 2.7
- CDF 3.0
- CDF 3.1
- HDF5 (*under consideration*)

KAMELEON Access/Interpolation Library



Call from any C supported Programming Language:

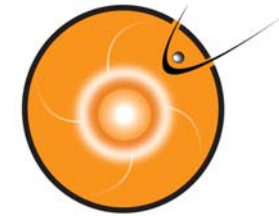
- Fortran
- C/C++
- IDL
- OpenDx
- Java
- Perl
- Vtk
- Your App

Current Standardized Model Output Availability

- BATSRUS
- OpenGGCM / UCLA-GGCM
- CTIP
- ENLIL
- MAS (*Beta Version*)

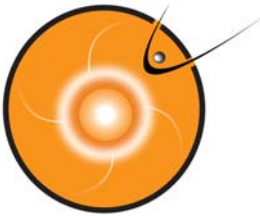
Currently Supported Science Data Formats

- CDF 2.7
- CDF 3.0
- CDF 3.1
- HDF5 (*under consideration*)



General Usage and Benefits

- 🌀 Self descriptive data files
- 🌀 Platform independent
- 🌀 Promotes data sharing
- 🌀 Speed and efficiency of direct data access
- 🌀 Same interface regardless of model/**data** input
- 🌀 Facilitates code reuse
- 🌀 Kameleon library allows model data to be more easily integrated into existing analysis and software applications
- 🌀 ... addresses some of the needs and requirements of “power-users”, as identified from the user feedback sessions from Monday.



Specific Usage and Benefits

CCMC:

- Runs-On-Request: Converted Data & Kameleon Download - to be automated
- CCMC Visualization: Space Weather Explorer
- CCMC Visualization: Space Weather View
- Particle Tracing
- Custom/Specialized Analysis Software: Field/Flow Line Tracing
- Derived Library Add-On for Kameleon -D. Berrios

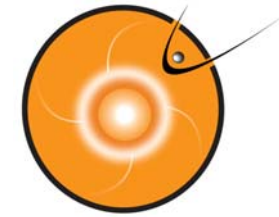
External Research & Analysis Packages:

- MAGIC - MAGnetogram Interpolation & Composition - Magnetogram Synthesis
- Themis Support
- Visbard Integration
- Possible integration with CISMDx Viz tool

Comparing Model data and observational data

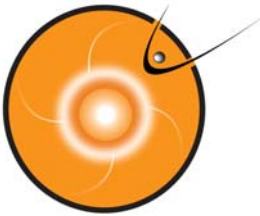
General data analysis - diverse set of users that have requested and used Kamaelon

Access/Interpolation library is highly configurable and expandable.



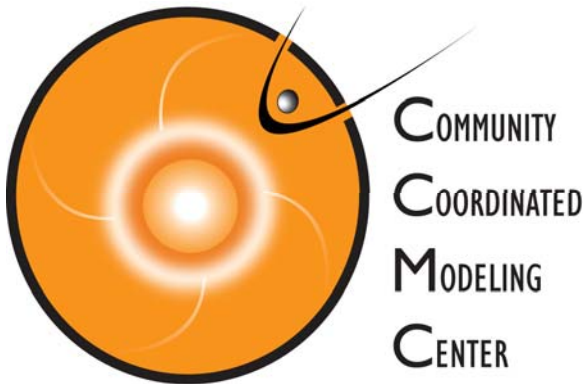
TODO List / Things To Remember

- ⦿ Variable naming conventions
- ⦿ Unit conversions
- ⦿ Grid description refinement
- ⦿ Coordinate transformations between native and target grid(s)
- ⦿ Opening Multiple files in memory with targeted interpolation on specific data set
- ⦿ Fulfilling expanded feature requests
- ⦿ Extracting Kamelon converter structure manager for external use
- ⦿ Refining access/interpolation library as feedback is acquired
- ⦿ Identifying an extensive list of desired/requested routines and functionality
- ⦿ Working with external groups - identifying methods to formally provide standardized model data along with the Kameeon access/interpolation library
- ⦿ Configuration Management: Model, Converter, Access Library, Container versions



Summary

- Metadata is a key component.
 - clearly defined set of core metadata elements that are currently being implemented on Kameleon converted data sets
 - Recently started collaborating with SPASE Working Group
- Structure oriented architecture of Kameleon Converter ensures flexibility expandability
 - Internal kameleon conversion functionality can ultimately be used by external developers
- Kameleon Software Suite currently supports:
 - BATSRUS, OpenGGCM, CTIP, ENLIL, & MAS
 - Select Observational data sets for MAGIC / Magnetogram Synthesis
- Kameleon access/interpolation library key features:
 - Interface to easily extract global & variable metadata
 - Time interpolation for MHD data sets
 - Fortran interface
 - IDL interface
 - Derived Library

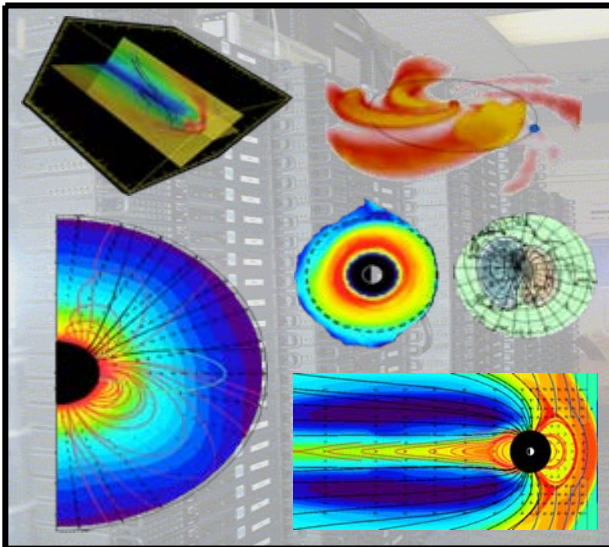


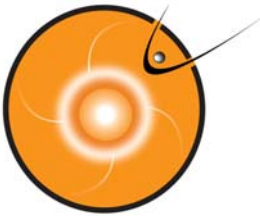
CCMC Data Standards Data Conversion, Access, & Interpolation Using Kameleon

Marlo Maddox

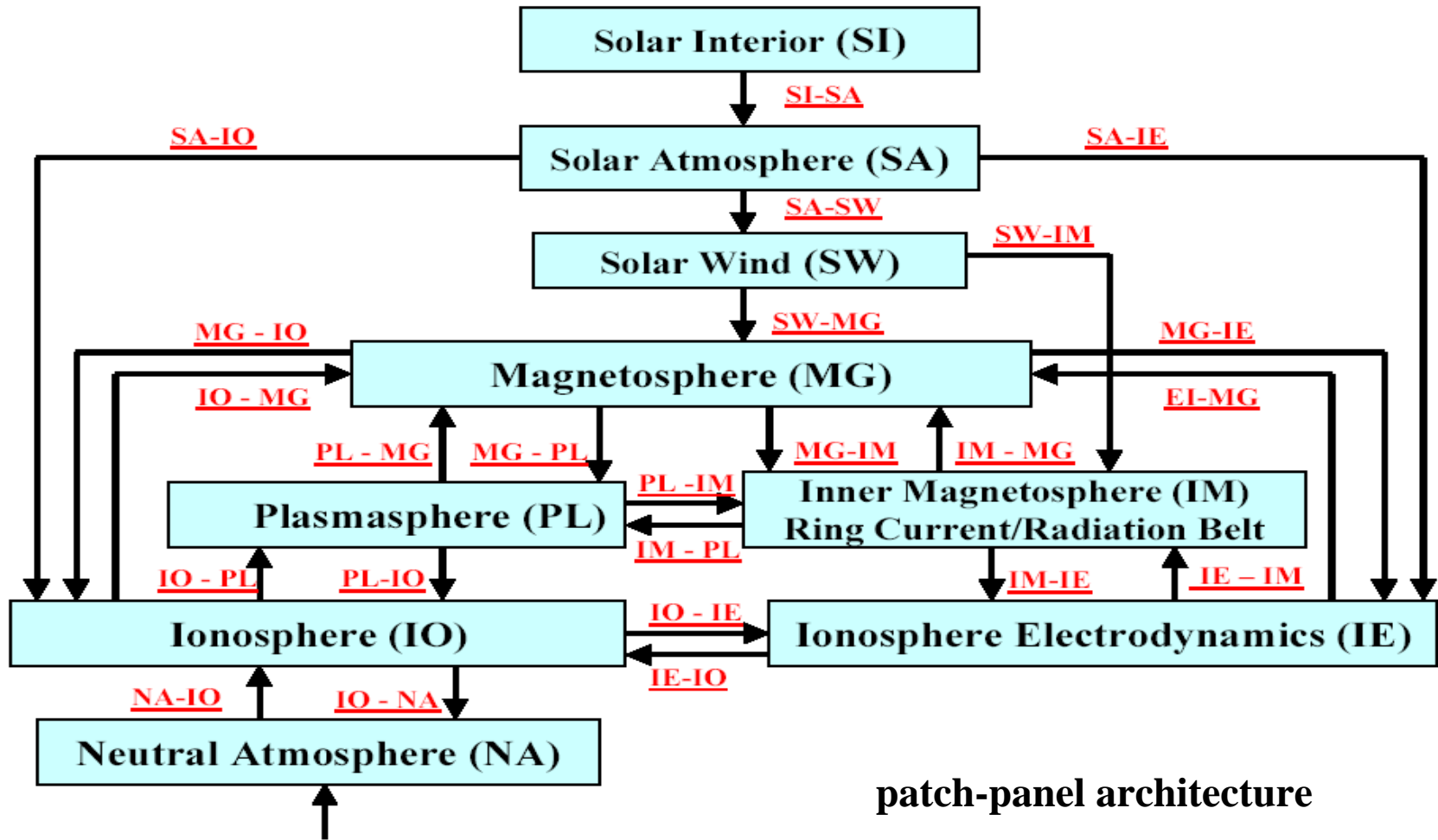
<http://ccmc.gsfc.nasa.gov>

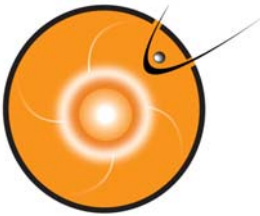
NASA Goddard Space Flight Center





Space Weather Models





Data Format Standard Options

- CDF
- HDF, HDF4, HDF5
- NetCDF
- FITS
- GRIB
- BUFR
- GRADS
- Office Note 29
- Office Note 84
- VICAR
- PDS
- Open Dx Data Model



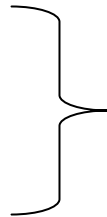
C example of 4D time interpolation of CCMC standardized data using access/interpolation library

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main( int argc, char *argv[] )
{
extern long init_time(char *, double *, double *);
extern float time_interpolate(char*,double,float,float,float);
long status;
char data_path[750];
char variable[10];
float X, Y, Z;
double time, start_time, end_time;
float sample_time_interval;
float time_interpolated_value;
```



variable declarations

```
strcpy( data_path, argv[1] );
strcpy( variable, argv[2] );
X = atof( argv[3] );
Y = atof( argv[4] );
Z = atof( argv[5] );
sample_time_interval = atof( argv[6] );
```



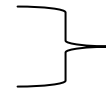
program input

```
status = init_time( data_path, &start_time, &end_time );
```



interpolation library call

```
printf("Simulation start_time:\t%f msec\n", start_time );
printf("Simulation end_time:\t%f msec\n", end_time );
```



stdout

```
for(time=start_time;time<=end_time;time+=sample_time_interval)
{
time_interpolated_value=time_interpolate(variable,time,X,Y,Z);
printf( "%s [ %f, %f, %f ] @ %f milliseconds\t%f\n",variable,
X,Y,Z,time,time_interpolated_value );
}
return 1;
}
```



interpolation library call

FORTRAN spatial interpolation example of CCMC standardized data using access/interpolation library

```
program f2c_interp_open_ggcm
c Three functions used to interpolate
c data from a specified batsrus cdf file
external f2c_open_cdf, f2c_close_cdf, f2c_interp_bats_cdf
c Variables to be used for interpolation and data extraction
character*150 cdf_file_path
real*8 x,y,z
real*8 interpolated_value
integer status
character*50 var_to_read
c --- set your actual path name here ---
cdf_file_path='open_ggcm.cdf '
c Open the cdf file
status=f2c_open_cdf(cdf_file_path)
c --- set your position values in GSE ---
x=-55.0
y=12.0
z=20.0
c --- set name of variable of interest ---
var_to_read='bx '
c --- call the interpolation routine ---
status=
1f2c_interp_open_ggcm_cdf(x,y,z,interpolated_value,var_to_read)
c --- close the currently open cdf file
status=f2c_close_cdf(0)
write(*,*) var_to_read, interpolated_value
end
```

variable declarations

Interpolation Library Call

library input

interpolation library calls