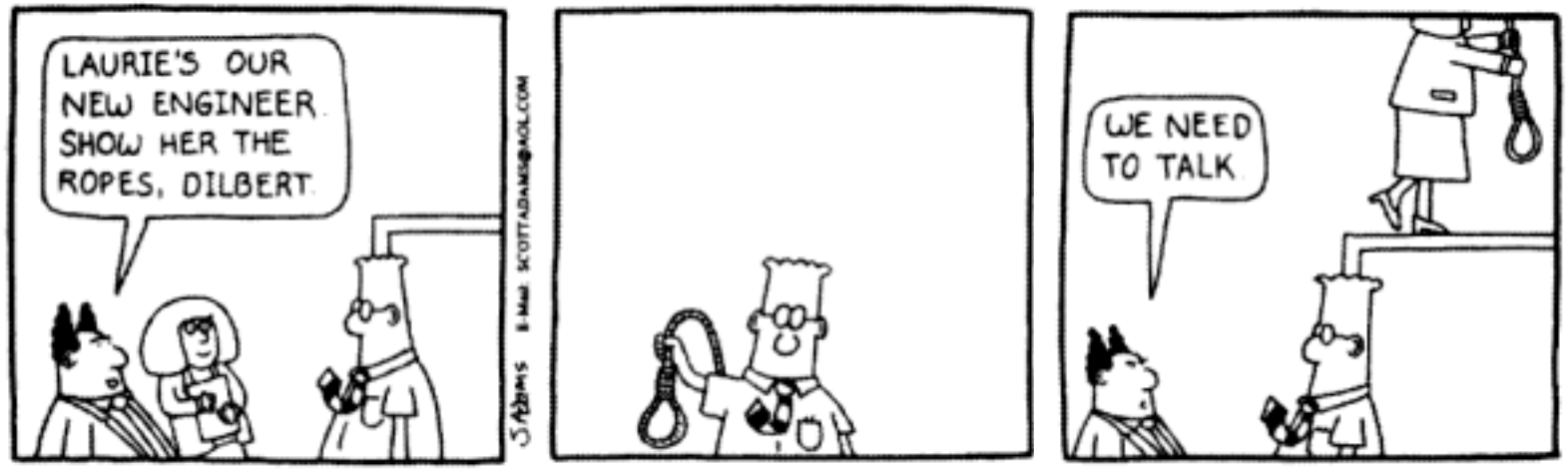
# LIS tutorial

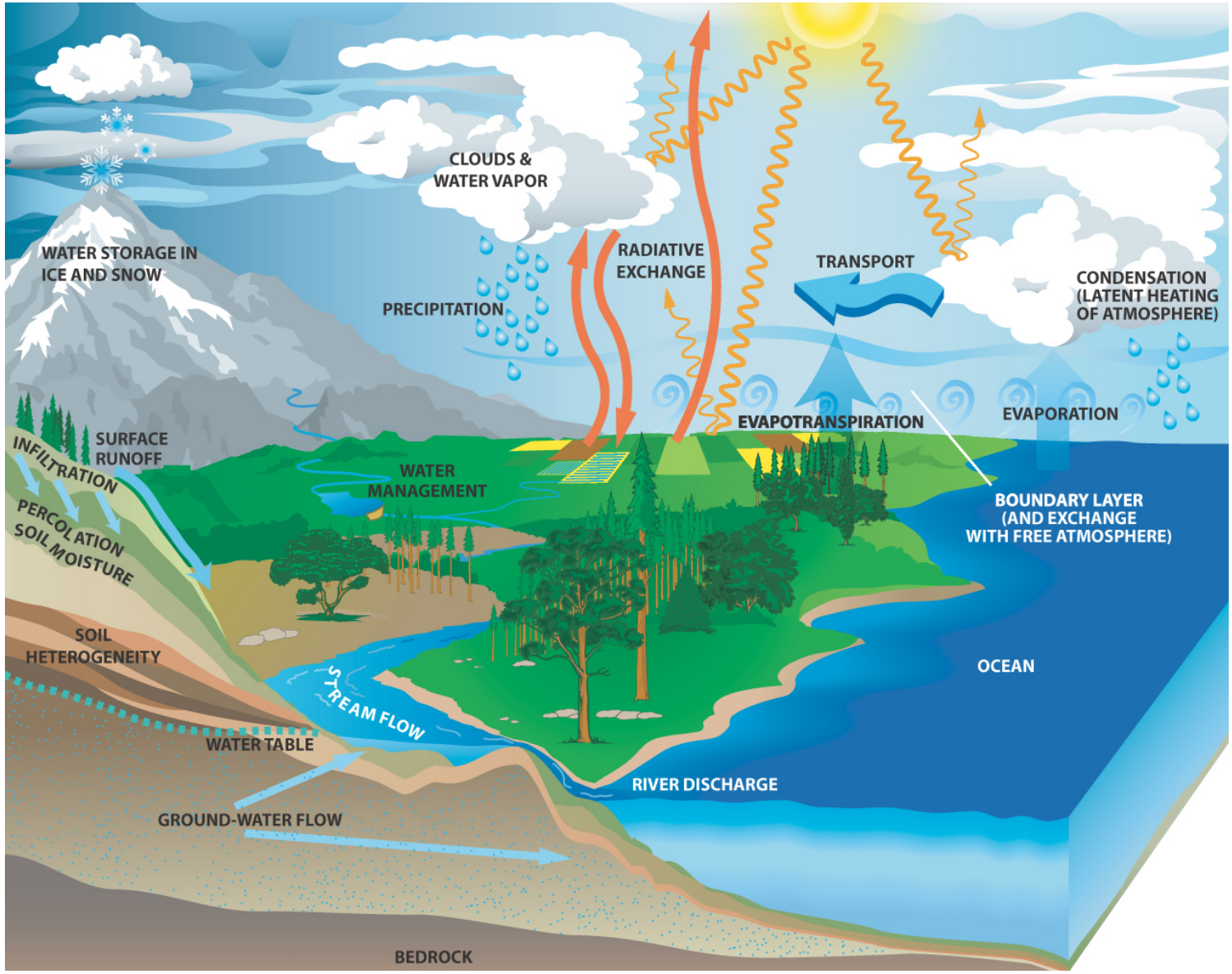
Sujay Kumar



# Outline

- Introduction
- Building LIS, using repository, code organization
- Software design philosophy, architecture
- Customizing LIS
- Visualizing LIS output

# Hydrological Cycle



# Hydrological Cycle

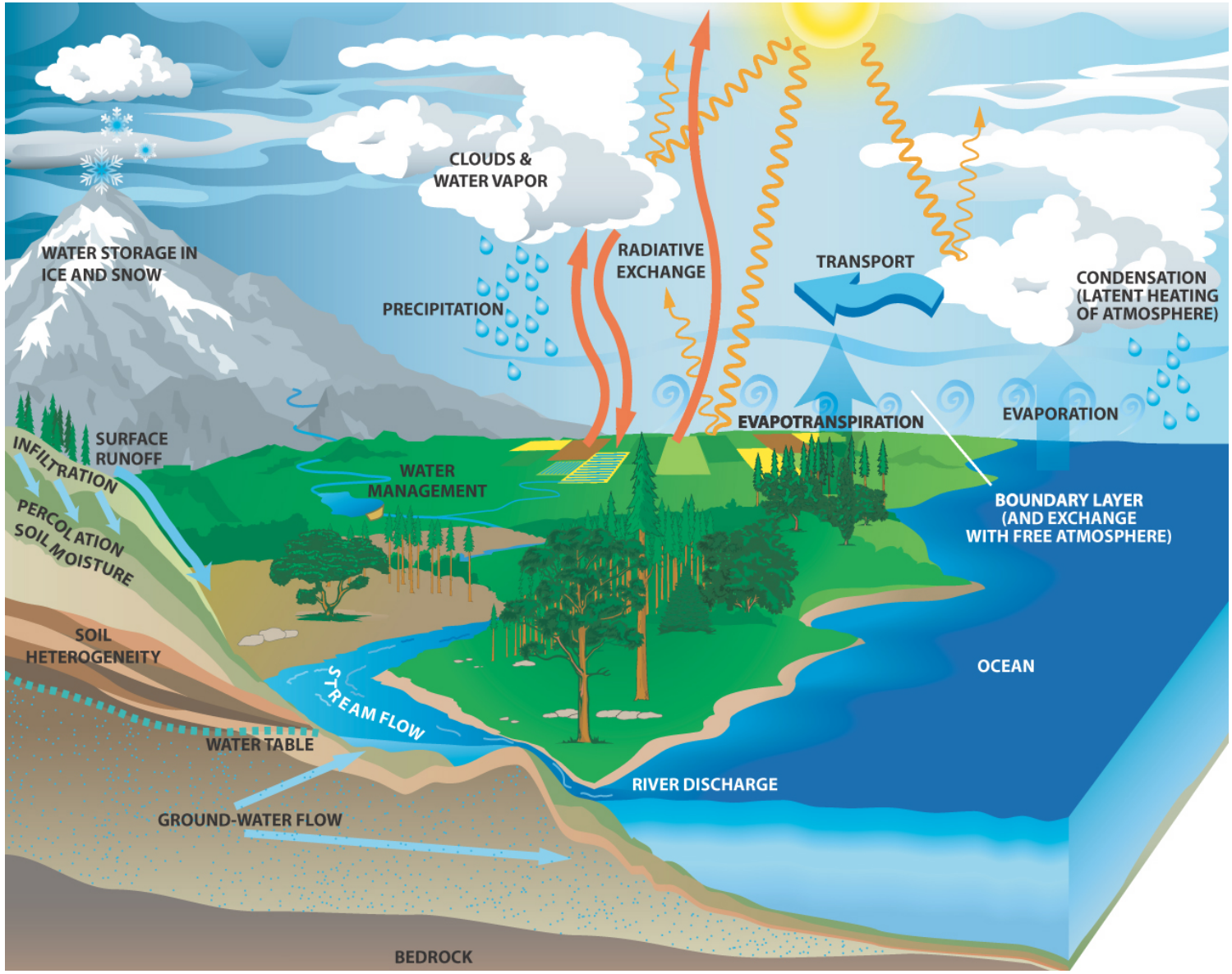
Snow and Ice

Evapotranspiration

Soil moisture

Surface water

Ground water



# Hydrological Cycle

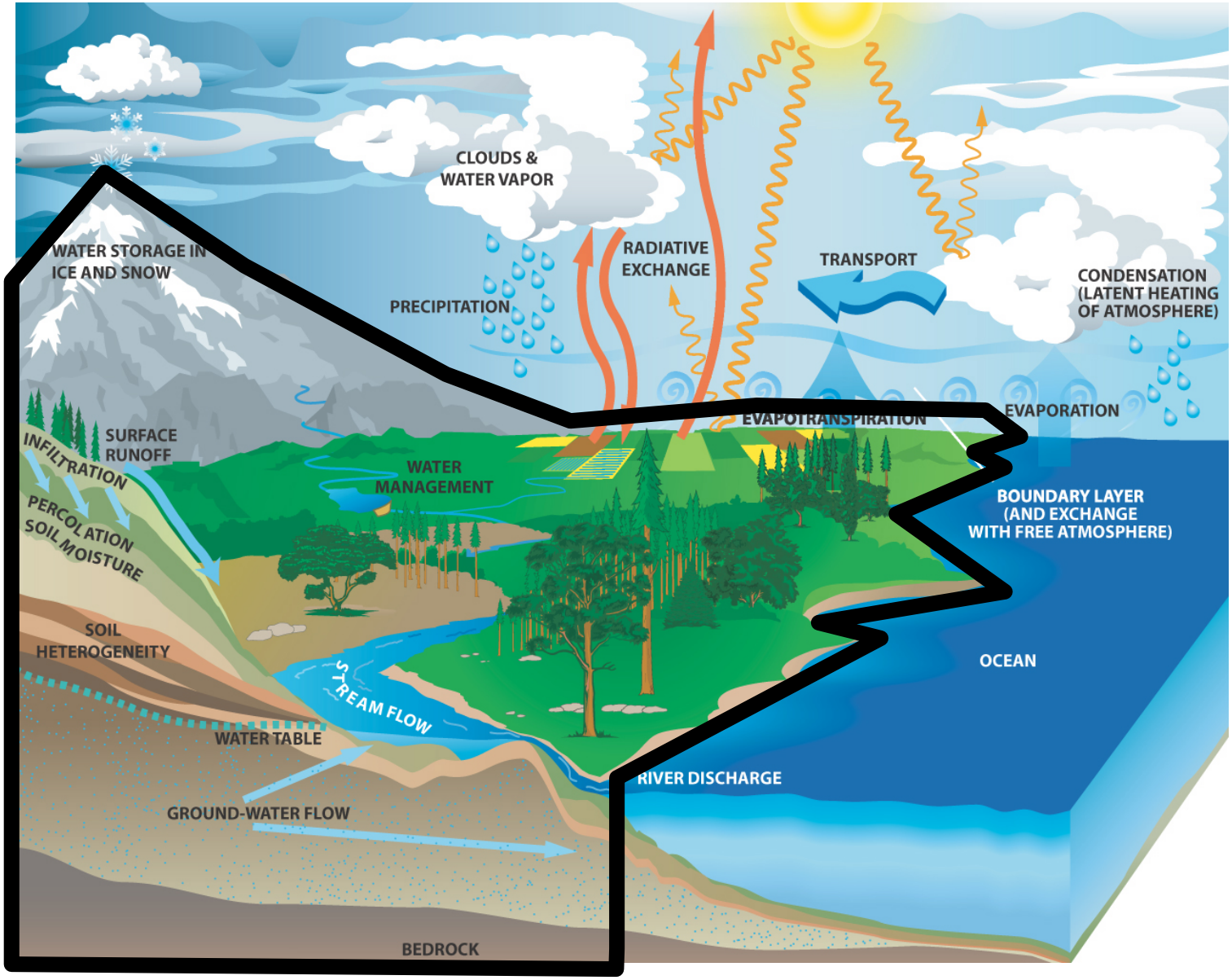
Snow and Ice

Evapotranspiration

Soil moisture

Surface water

Ground water



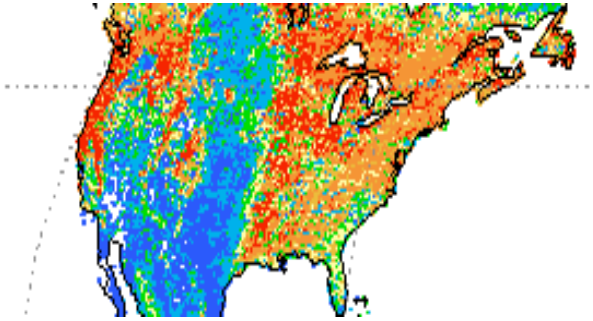
# LIS - heritage

- LIS is a land surface modeling and data assimilation system (LDAS)
- Capable of modeling at different spatial scales, globally and regionally

**Kumar et al. (2006): Land Information System: An interoperable Framework for High Resolution Land Surface Modeling, Environmental Modeling and Software, Vol 21, pp 1402-1415.**

# LIS - heritage

- LIS is a land surface modeling and data assimilation system (LDAS)
- Capable of modeling at different spatial scales, globally and regionally

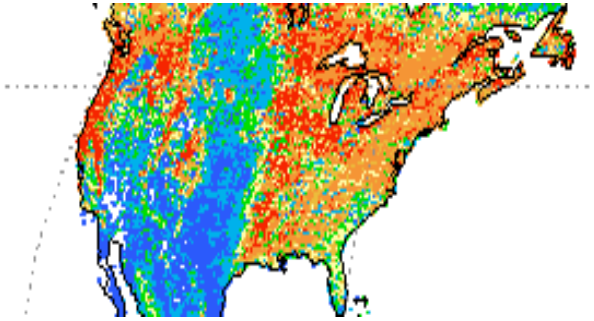


North American LDAS
1/8th degree spatial
resolution

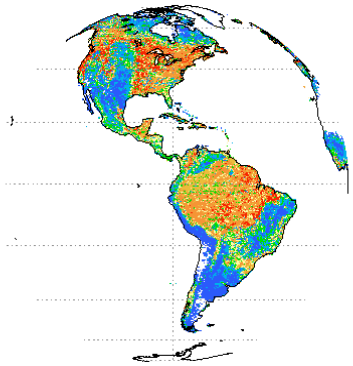
**Kumar et al. (2006): Land Information System: An interoperable Framework for High Resolution Land Surface Modeling, Environmental Modeling and Software, Vol 21, pp 1402-1415.**

# LIS - heritage

- LIS is a land surface modeling and data assimilation system (LDAS)
- Capable of modeling at different spatial scales, globally and regionally



North American LDAS
1/8th degree spatial resolution

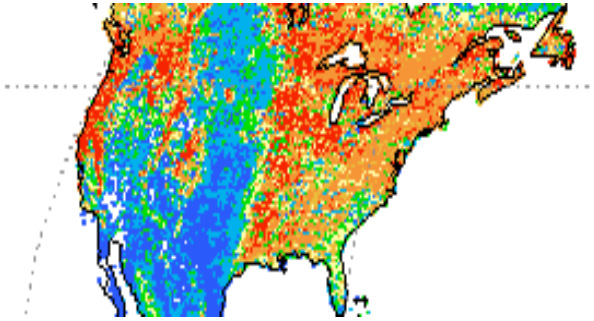


Global LDAS
1/4th degree spatial resolution

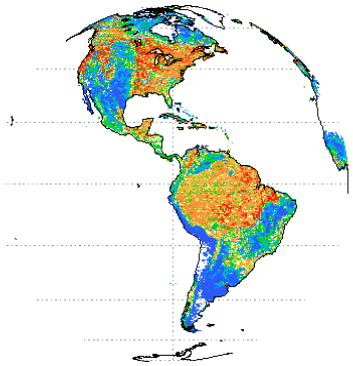
**Kumar et al. (2006): Land Information System: An interoperable Framework for High Resolution Land Surface Modeling, Environmental Modeling and Software, Vol 21, pp 1402-1415.**

# LIS - heritage

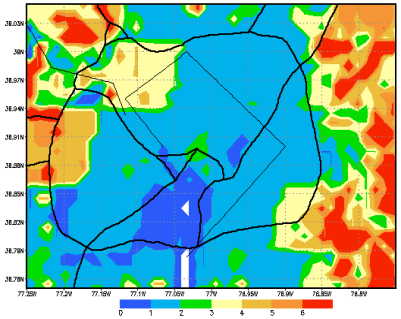
- LIS is a land surface modeling and data assimilation system (LDAS)
- Capable of modeling at different spatial scales, globally and regionally



North American LDAS
1/8th degree spatial resolution



Global LDAS
1/4th degree spatial resolution



LIS
global, regional, point
upto 1km and finer

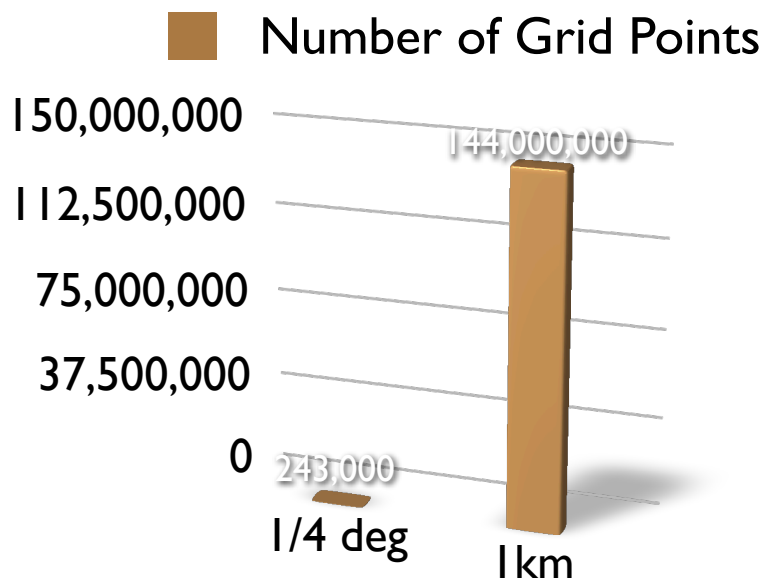
**Kumar et al. (2006): Land Information System: An interoperable Framework for High Resolution Land Surface Modeling, Environmental Modeling and Software, Vol 21, pp 1402-1415.**

# More History

- Original Goal : to Enable Global Land Surface Modeling at 1km spatial resolution
- Huge Computational Challenge

# More History

- Original Goal : to Enable Global Land Surface Modeling at 1km spatial resolution
- Huge Computational Challenge

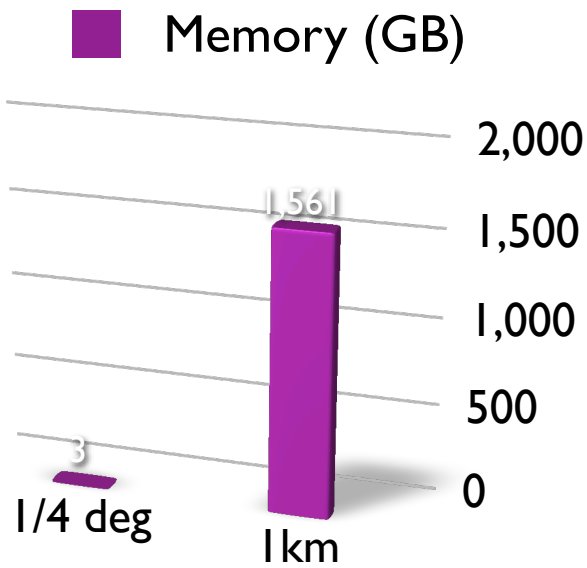


# More History

- Original Goal : to Enable Global Land Surface Modeling at 1km spatial resolution
- Huge Computational Challenge

# More History

- Original Goal : to Enable Global Land Surface Modeling at 1km spatial resolution
- Huge Computational Challenge

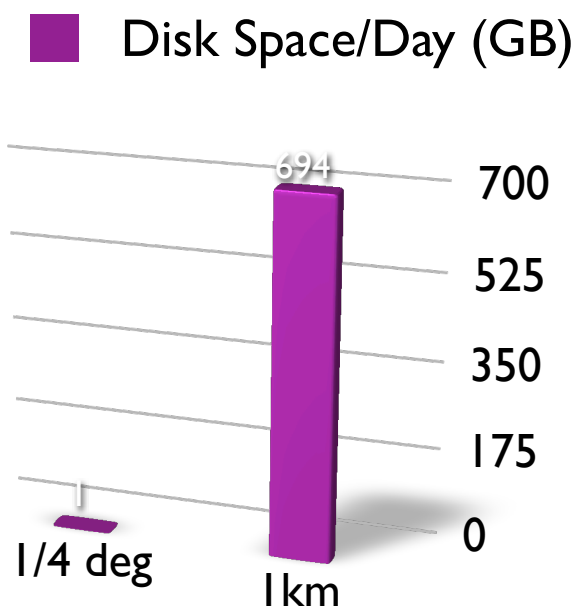


# More History

- Original Goal : to Enable Global Land Surface Modeling at 1km spatial resolution
- Huge Computational Challenge

# More History

- Original Goal : to Enable Global Land Surface Modeling at 1km spatial resolution
- Huge Computational Challenge

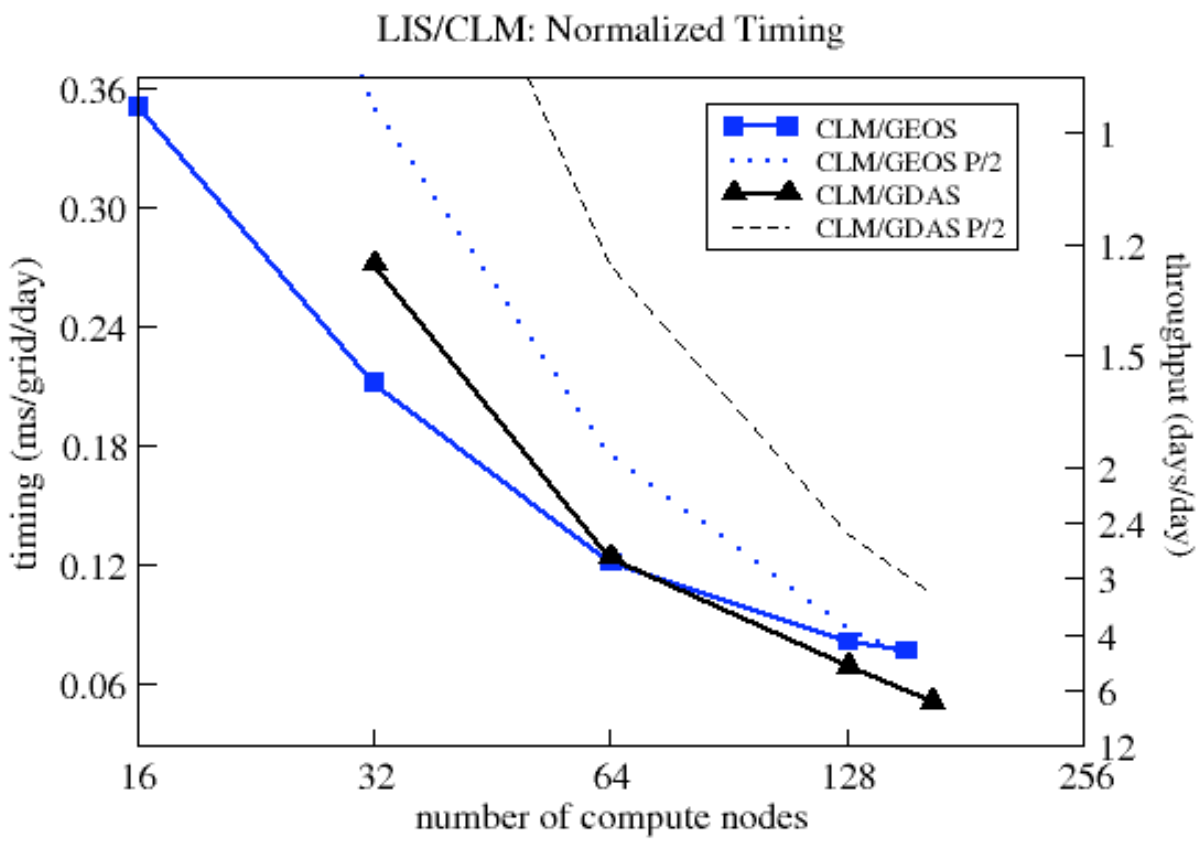


# More History

- Original Goal : to Enable Global Land Surface Modeling at 1km spatial resolution
- Huge Computational Challenge

# More History

- Original Goal : to Enable Global Land Surface Modeling at 1km spatial resolution
- Huge Computational Challenge



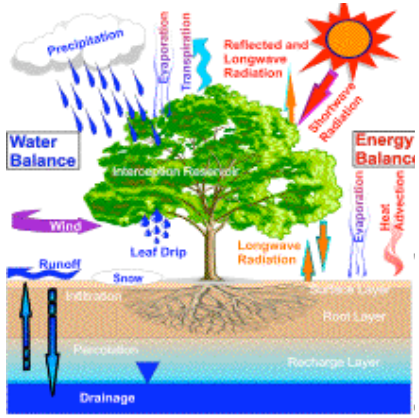
# Program Flow

Uncoupled or Analysis Mode

Parameters (Topography, Soil properties, vegetation properties)

Meteorological Boundary Conditions (Forcings)

Land Surface Models (Noah, CLM, Catchment, Mosaic, HySSIB, SiB2, Sacramento, SNOW17)




Water and Energy Fluxes, Soil Moisture and Temperature profiles, Land surface states

# Program Flow

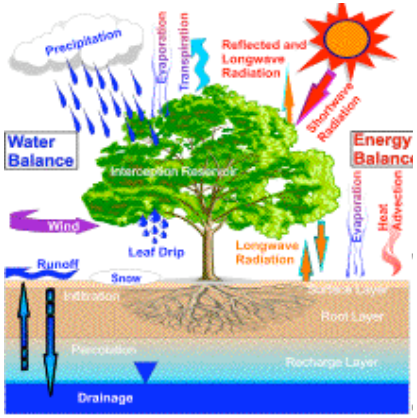
Uncoupled or Analysis Mode

Parameters (Topography, Soil properties, vegetation properties)

Meteorological Boundary Conditions (Forcings)

Observations (Soil Moisture, Snow, Skin Temperature)

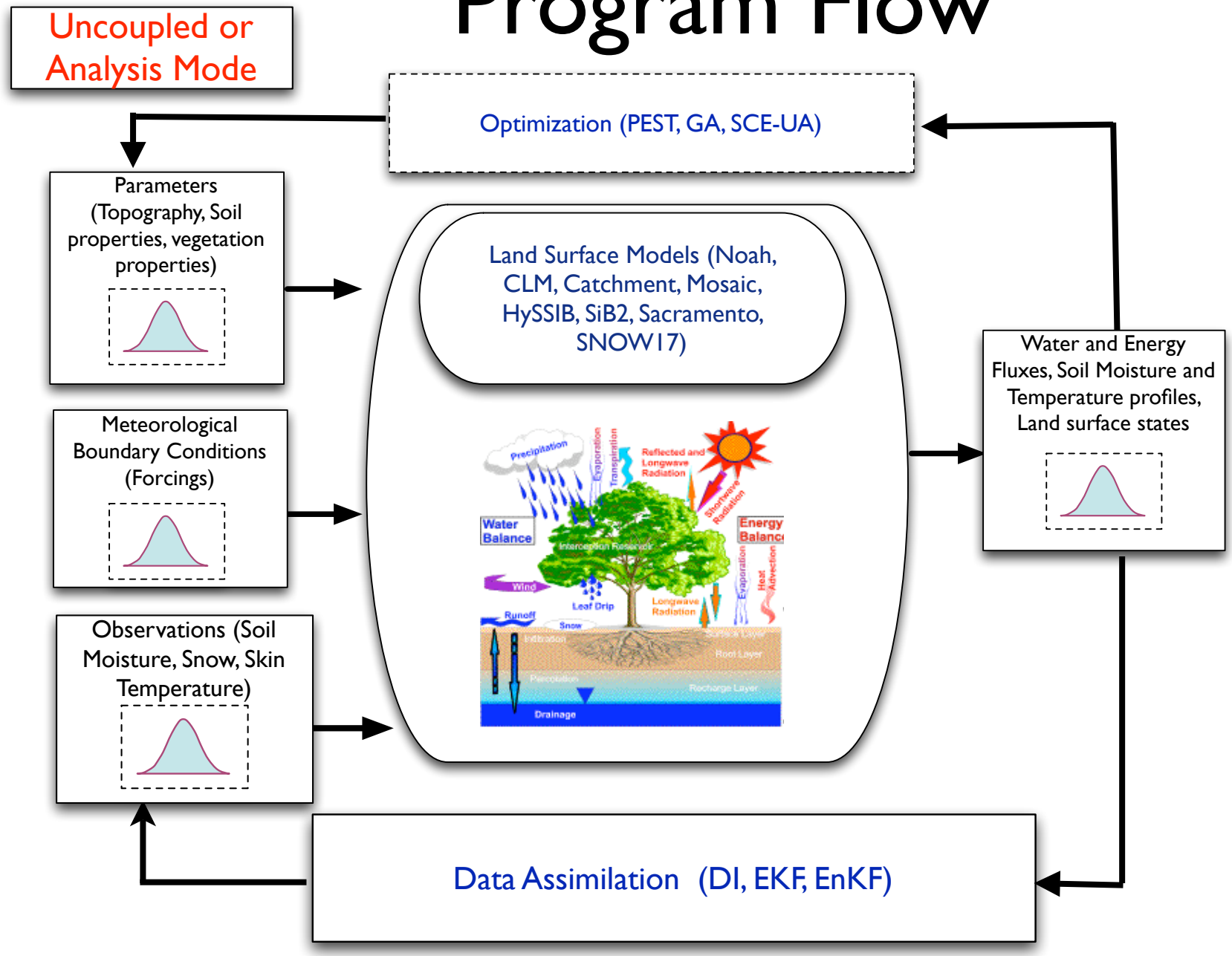
Land Surface Models (Noah, CLM, Catchment, Mosaic, HySSIB, SiB2, Sacramento, SNOW17)




Water and Energy Fluxes, Soil Moisture and Temperature profiles, Land surface states

Data Assimilation (DI, EKF, EnKF)

# Program Flow



# Program Flow

Uncoupled or Analysis Mode

Coupled or Forecast Mode

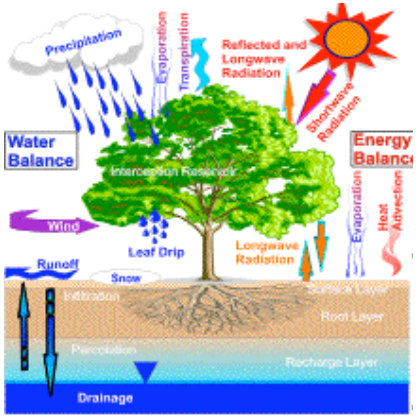
Optimization (PEST, GA, SCE-UA)

Parameters (Topography, Soil properties, vegetation properties)

Meteorological Boundary Conditions (Forcings)

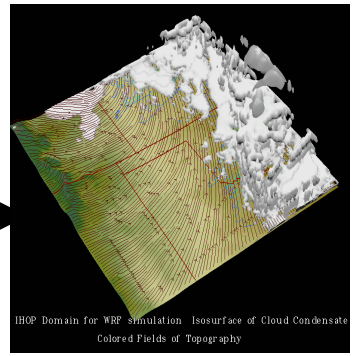
Observations (Soil Moisture, Snow, Skin Temperature)

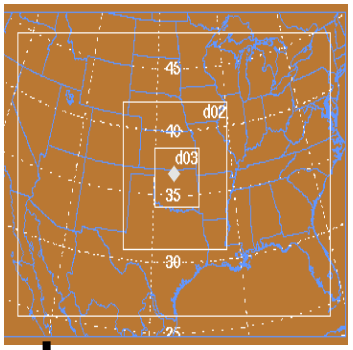
Land Surface Models (Noah, CLM, Catchment, Mosaic, HySSIB, SiB2, Sacramento, SNOW17)




Water and Energy Fluxes, Soil Moisture and Temperature profiles, Land surface states

LIS - WRF Interface




WRF

Data Assimilation (DI, EKF, EnKF)

# LIS source code

http:://lis.gsfc.nasa.gov

http:://modelingguru.nasa.gov

NASA :: Goddard Space Flight Center : Land Information Systems

http://lis.gsfc.nasa.gov/

Goddard Space Flight Center

Land Information System

Home
Overview
Publications
Documentation

The award-winning Land Information System (LIS) is a high performance land surface modeling and data assimilation system, based on GSFC's Land Data Assimilation Systems. This site provides LIS source codes and sample data to registered users.

LIS 5.0-beta Public Release Home Page

http://lis.gsfc.nasa.gov/Source/LIS5.0-BETA/index.html

Land Information System 5.0-beta (LIS 5.0)

Public Release Home Page

Documentation, Source Code, and Input and Output Data

This web page contains links to documentation, source code, and input data for the version 5.0-beta release of LIS.

Model Documentation

NASA Modeling Guru: Clearspace: NASA Modeling Guru

https://modelingguru.nasa.gov/

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

+ NASA Portal
+ Request an Account
+ Modeling Guru Home

Modeling Guru
Beta

Welcome, Guest Login

Enter Search Term

NASA Modeling Guru

Modeling Guru is a research and collaboration resource for all those concerned with NASA scientific models or NASA's High End Computing (HEC) systems.

New to Community? Close

Take a quick tour to learn how things are organized and how to best use Clearspace to work with your team.

Communities

- Space Science Models
  - DYNAMO
- Land & Atmospheric Dynamics Models
  - Cirrus
  - fvCore
  - GCE
  - GEOS
  - GEOS-DAS
  - GISS ModelE
  - Land Information System
- Atmospheric Chemistry Models
  - GEOS-CHEM
  - GMI
- Ocean Models
  - Poseidon

Welcome NASA Modelers

**Modeling Guru**'s combined forums and knowledge base is becoming a repository for the accumulated expertise of NASA's scientific modeling community. All NASA modelers and associates are encouraged to participate and help us fill the "holes" in our modeling knowledge.

**Disclaimer:** While this is a government resource, it is not totally screened for content. By accessing this site, you agree to use it at your own risk.

Become a Registered Member Today!

Registration offers you a number of benefits, including being able ask questions, contribute your expertise, and receive email notifications. View registration instructions in the Modeling Guru Registration FAQ.

New to Modeling Guru?

This site offers a number of powerful features, including "wiki" documents, RSS feeds, search tools, and more. For a brief intro, see Modeling Guru Basics or take the more-detailed Quick Tour.

Important Links

- How To Register
- Frequently Asked Questions
- Posting Policy

What's New

Go to: Discussions Documents Blog Posts

Land Information System

Overview | All Content (30) | Discussions (25) | Documents (5)

Welcome to the LIS community!

The Land Information System (LIS; http://lis.gsfc.nasa.gov; Kumar et al (2006), Peters-Lidard et al. (2007) is a land surface modeling and data assimilation system developed at NASA GSFC that integrates the use of land surface models, high resolution satellite and observational data, data assimilation techniques and high performance computing tools. LIS operates primarily on an ensemble of land surface models over user-specified regional and global domains. The LIS software is designed using object oriented design principles so that a variety of typical land modeling and assimilation functions are abstracted to function in an interoperable manner. A key new functionality in LIS is the support for sequential data assimilation extensions, enabling the interoperable use of multiple observational sources, land surface models, and data assimilation algorithms (Kumar et al. 2008).

Please acknowledge the use of LIS in your publications using the following LIS-related references:

Kumar, S. V., C. D. Peters-Lidard, Y. Tian, P. R. Houser, J. Geiger, S. Olden, L. Lighty, J. L. Eastman, B. Doty, P. Dirmeyer, J. Adams, K. Mitchell, E. F. Wood and J. Sheffield, 2006. Land Information System - An Interoperable Framework for High Resolution Land Surface Modeling. Environmental Modelling & Software, Vol. 21, 1402-1415.

Peters-Lidard, C. D., P. R. Houser, Y. Tian, S. V. Kumar, J. Geiger, S. Olden, L. Lighty, B. Doty, P. Dirmeyer, J. Adams, K. Mitchell, E. F. Wood and J. Sheffield, 2007. High-performance Earth system modeling with NASA/GSFC's Land Information System. Innovations in Systems and Software Engineering . Vol. 3(3), 157-165.

Kumar, S.V., R.H. Reichle, C. D. Peters-Lidard, R. D. Koster, X. Zhan, W. T. Crow, J. B. Eylander, and P. R. Houser: A Land Surface Data Assimilation Framework using the Land Information System: Description and Applications, Advances in Water Resources, 2007, in print.

Recent Content

| Author | Subject | Views | Replies | Last Activity |
|---|---|---|---|---|
| wd20wz | run_domain: T382 Gaussian Projection | 26 | 1 | 5 days ago by geiger |
| hirokok | suppforcing/cmap | 46 | 2 | 1 week ago by hirokok |
| sujay | LISGCE users' guide | 13 | 0 | 1 week ago by sujay |

Recent Blog Posts

You can't post to a blog here, but watch blog po for new insights and ide

Actions

Notifications
View RSS feeds

Tag Cloud

lis data correction dom elevation forcing grid land nldas suppforcing #inc_water_points coupled netcdf sgi userguide

View all

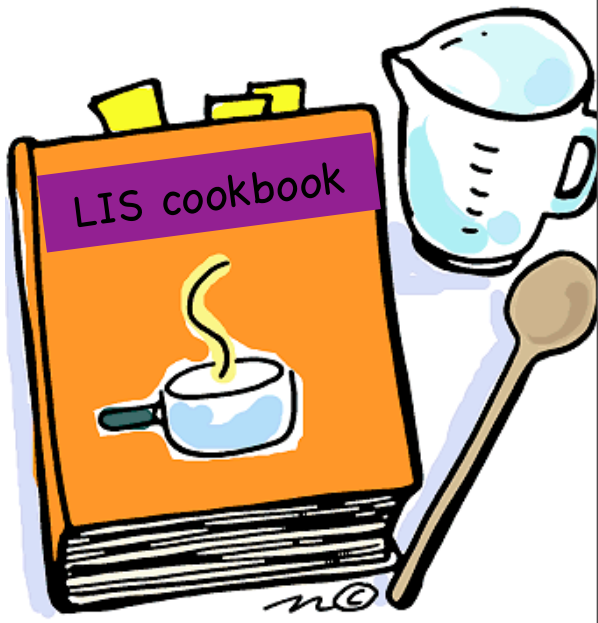
Top Members

sujay
geiger
fan
jonc
cbblanke

View all participants

# Software Requirements

- ☑ Fortran 90/95 compiler (g95 will not work for LIS5.0)
  - ☐ preferred : intel, pgi, lahey, absoft
- ☑ C compiler
- ☐ MPI - if parallel processing capability is desired
- ☑ Earth System Modeling Framework (ESMF)
  - ☐ 2.2.2rp3 - for LIS5.0
  - ☐ 3.1.0r - for LIS 6.0
- ☐ LIS supports Grib1, NETCDF, HDF formats



# LIS Documentation

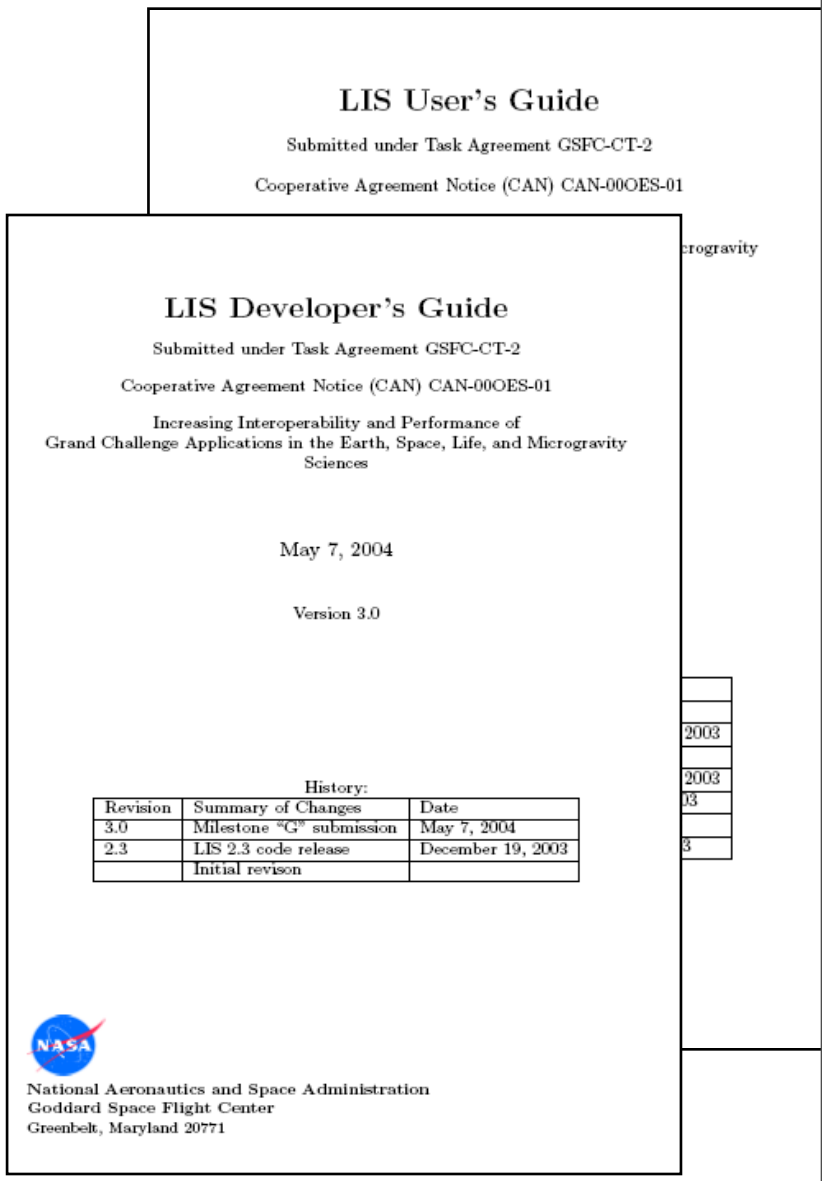
- User's guide

  Step-by-step instructions on how to build the LIS code

- Developer's Guide

  Instructions on how to bring in new functionalities (LSMs, forcing schemes, Data Assimilation, parameter data, etc.)

- Reference Manual



LIS User's Guide

Submitted under Task Agreement GSFC-CT-2

Cooperative Agreement Notice (CAN) CAN-00OES-01

LIS Developer's Guide

Submitted under Task Agreement GSFC-CT-2

Cooperative Agreement Notice (CAN) CAN-00OES-01

Increasing Interoperability and Performance of
Grand Challenge Applications in the Earth, Space, Life, and Microgravity Sciences

May 7, 2004

Version 3.0

History:

| Revision | Summary of Changes | Date |
|---|---|---|
| 3.0 | Milestone "G" submission | May 7, 2004 |
| 2.3 | LIS 2.3 code release | December 19, 2003 |
| | Initial revison | |

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

# Getting LIS source

- Use the subversion repository (https://flood.gsfc.nasa.gov)
- Apply for an account (James.V.Geiger@nasa.gov)
- Request a "Project Release" of the LIS code (http://lis.gsfc.nasa.gov/register.shtml)
- Check out the code

# LIS source code repository

- Helpful links

  http://subversion.tigris.org/

  http://svnbook.red-bean.com/

- Check out the LIS code:

  svn co https://flood.gsfc.nasa.gov/svn/5/public/ src

# LIS source code repository

- Helpful links

  http://subversion.tigris.org/

  http://svnbook.red-bean.com/

- Check out the LIS code:

  svn co https://flood.gsfc.nasa.gov/svn/5/public/ src

# LIS source code repository

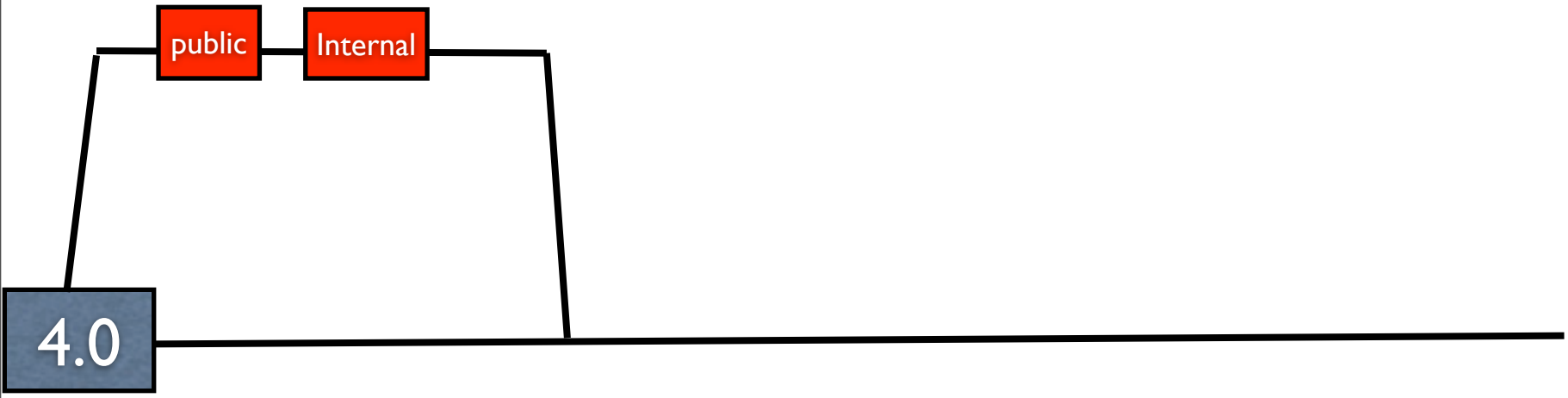
- Helpful links

  http://subversion.tigris.org/

  http://svnbook.red-bean.com/

- Check out the LIS code:

  svn co https://flood.gsfc.nasa.gov/svn/5/public/ src



# LIS source code repository

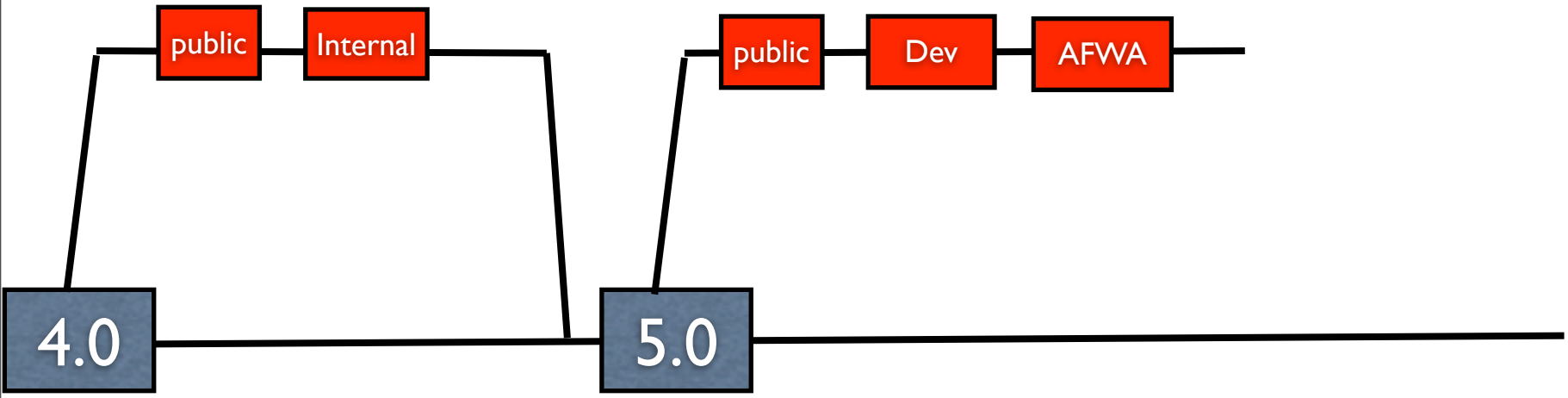
- Helpful links

  http://subversion.tigris.org/

  http://svnbook.red-bean.com/

- Check out the LIS code:

  svn co https://flood.gsfc.nasa.gov/svn/5/public/ src



# LIS source code repository

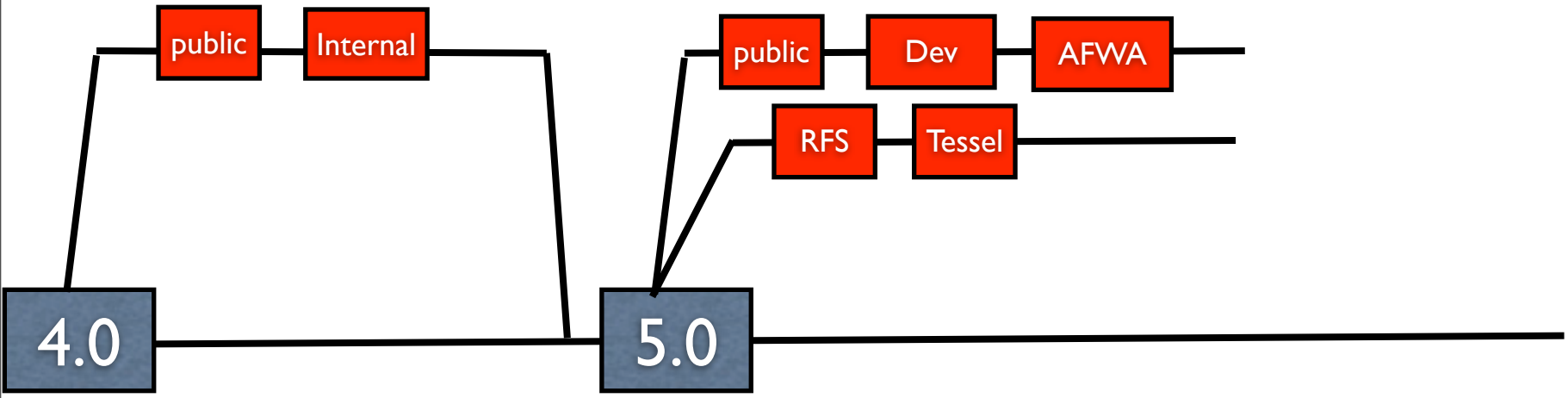
- Helpful links

  http://subversion.tigris.org/

  http://svnbook.red-bean.com/

- Check out the LIS code:

  svn co https://flood.gsfc.nasa.gov/svn/5/public/ src



# LIS source code repository

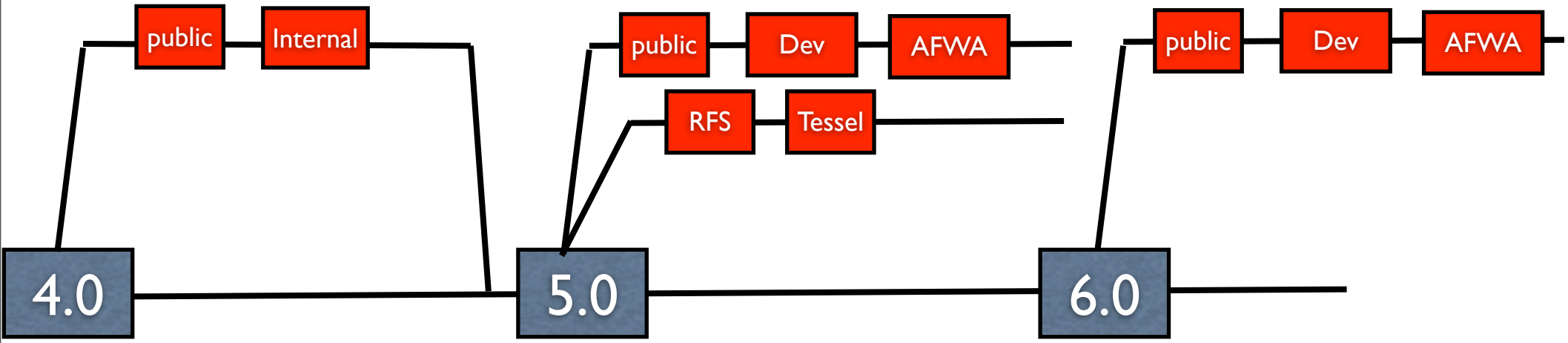
- Helpful links

  http://subversion.tigris.org/

  http://svnbook.red-bean.com/

- Check out the LIS code:

  svn co https://flood.gsfc.nasa.gov/svn/5/public/ src



# LIS source code repository

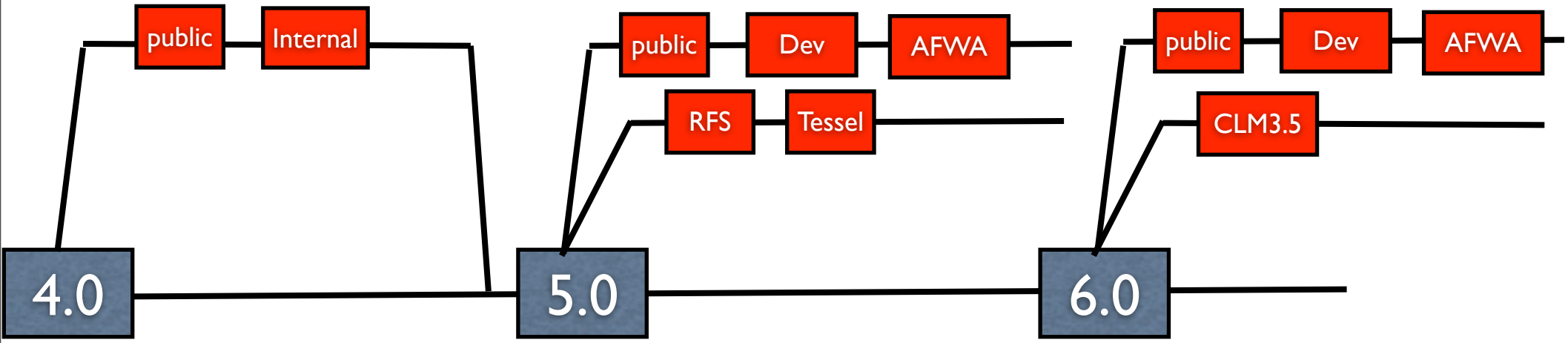
- Helpful links

  http://subversion.tigris.org/

  http://svnbook.red-bean.com/

- Check out the LIS code:

  svn co https://flood.gsfc.nasa.gov/svn/5/public/ src



# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

arch

- configure.lis.aix
- configure.lis.irix64
- configure.lis.linux_32_lf95
- configure.lis.linux_absoft
- configure.lis.linux_ifc
- configure.lis.linux_pgi
- configure.lis.macosx_gfortran
- configure.lis.macosx_intel
- configure.lis.osf1
- configure.lis.osf1.baselibs

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

- baseforcing
  - afwa
  - ecmwf
  - ecmwfreanal
  - gdas
  - geos
  - gmao_gldas
  - gswp
  - gswp1
  - princeton
  - rhone
  - template

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

- dataassim
  - algorithm
    - di
    - gmaoenkf
  - biasEstimation
    - gmaoBE
  - obs
    - AMSREsm
    - ISSCP_Tskin
    - MODISsca
    - multisynsmobs
    - SNODEP
    - syntheticlst1
    - syntheticsm1
    - syntheticsm2
    - syntheticswe1
  - perturb
    - gmaopert
    - static

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

# Code organization

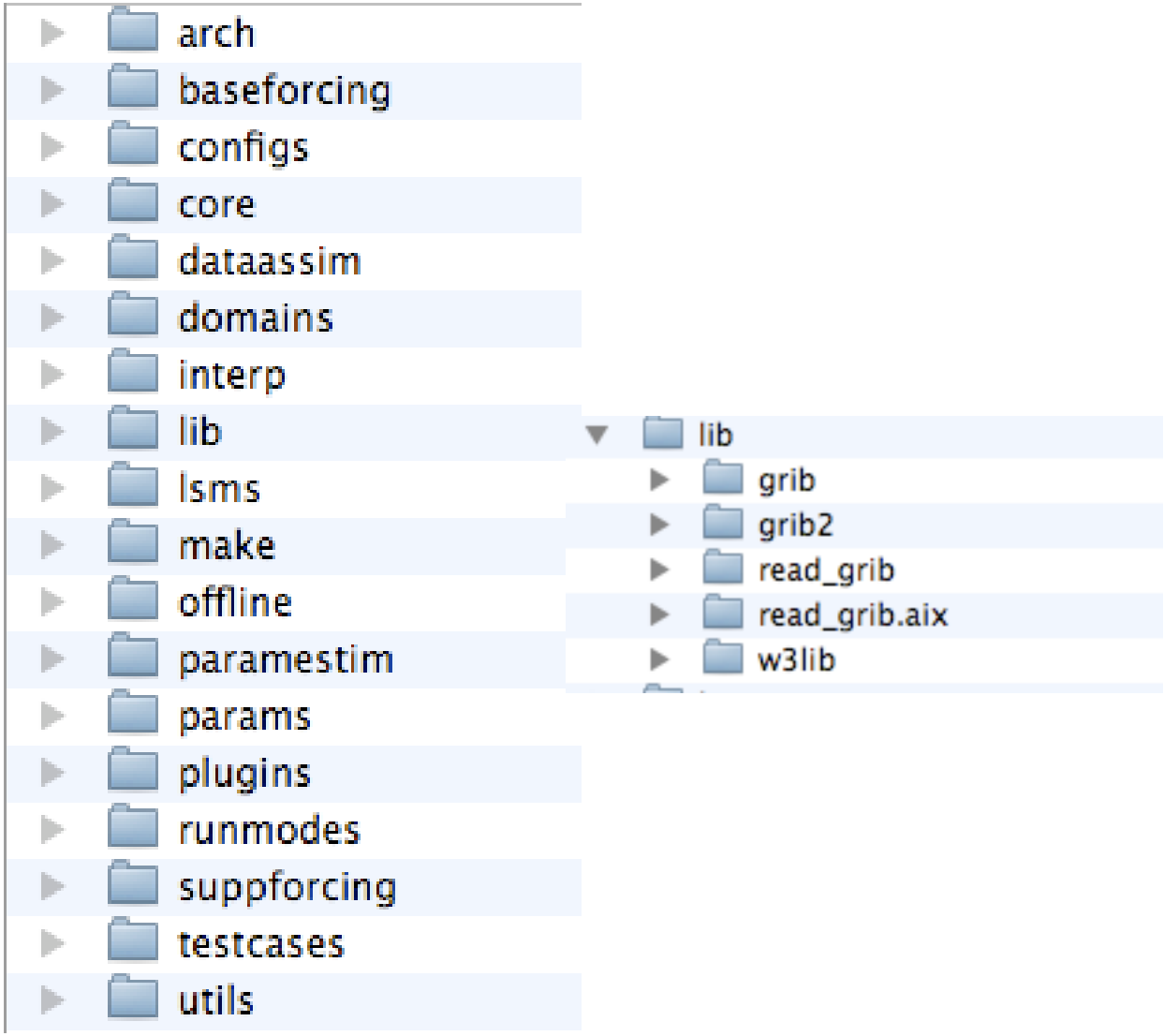
- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

- domains
  - afwa
  - catchment
  - gaussian
  - gswp
  - hrap
  - lambert
  - latlon
  - merc
  - polar
  - psglobal

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

# Code organization



# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

# Code organization

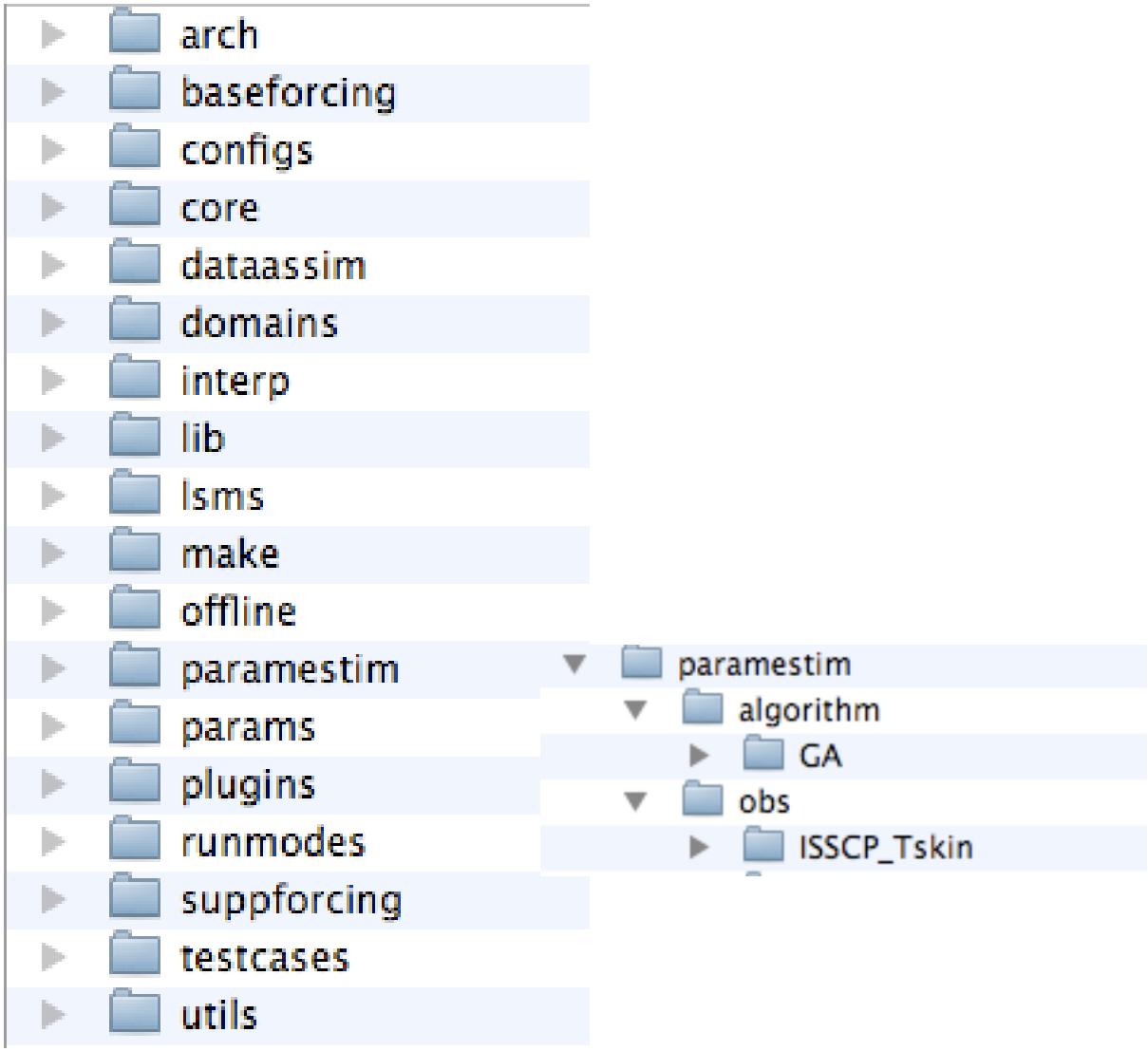
- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

- lsms
  - catchment
  - clm2
  - clm3.5
  - clm3.5.old
  - csu
  - hyssib
  - mlbc
  - mosaic
  - noah.2.7.1
  - noah.3.0
  - noah.afwa
  - sacramento
  - sacsnow17
  - SiB2
  - snow17
  - template
  - tessel
  - vic

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

# Code organization



# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

- params
  - albedo
  - gfrac
  - lai
  - landcover
  - soils
  - tbot
  - topo

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

- runmodes
  - agrmetmode
  - gce_cpl_mode
  - paramEstimation
  - retrospective
  - wrf_cpl_mode

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

- suppforcing
  - 3B42RT
  - 3B42V6
  - agrrad
  - ceop
  - cmap
  - cmorph
  - d2pcpcar
  - d2pcpokl
  - nldas
  - nldas2
  - saldas
  - scan
  - stg2
  - stg4

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

- testcases
  - arch
  - baseforcing
  - dataassim
  - domains
  - lib
  - lsms
  - misc
  - suppforcing

# Code organization

- arch
- baseforcing
- configs
- core
- dataassim
- domains
- interp
- lib
- lsms
- make
- offline
- paramestim
- params
- plugins
- runmodes
- suppforcing
- testcases
- utils

# Compiling LIS code

1. Build the libraries

   Grib1 (w3lib, read_grib, grib) - provided with the source code (under src/lib)

   ESMF - download the required version from http://www.esmf.ucar.edu

2. Build the dependency generator

   src/make/MAKDEP

3. Specify the configure.lis file that contains the architecture/compiler specific flags

   Sample files under src/arch

4. Edit the misc.h file

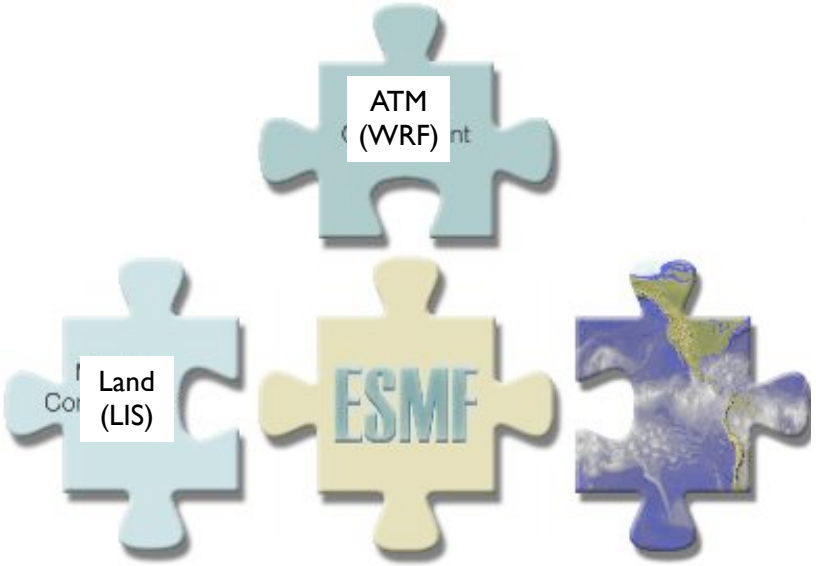
   to turn on/off parallel processing, NETCDF, HDF

   include water points or mask them

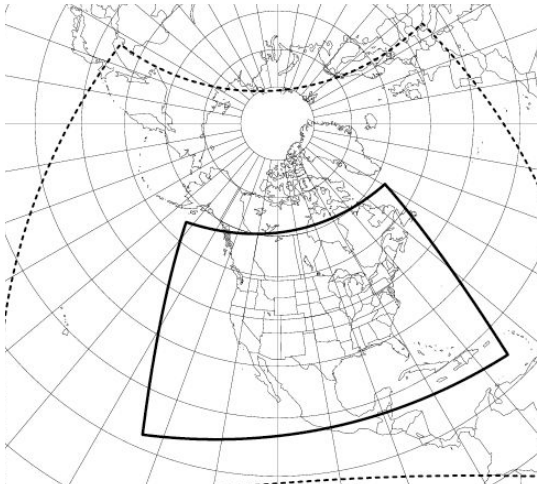
5. Finally, run gmake in src/make directory

# What is ESMF?

- Software for building and coupling weather, climate and related models
- Provides representations of Earth system grids, tools for mapping between them in multiprocessor environment
- Includes toolkits for building applications: time manager, error handling, resource management, parallel communications
- An application once it is wrapped with ESMF is known as a "Gridded component"
- Gridded components are coupled using "coupler components"

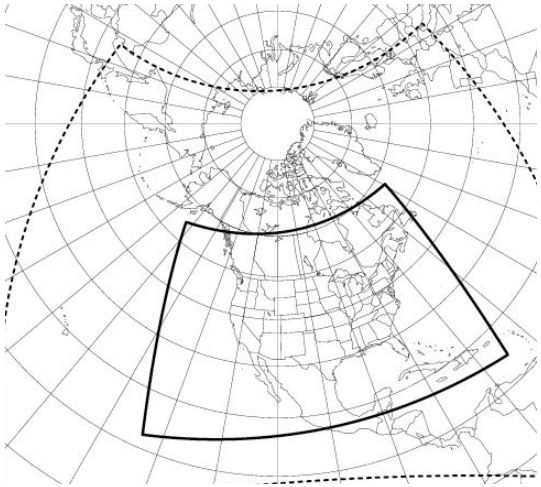


# Key ESMF objects



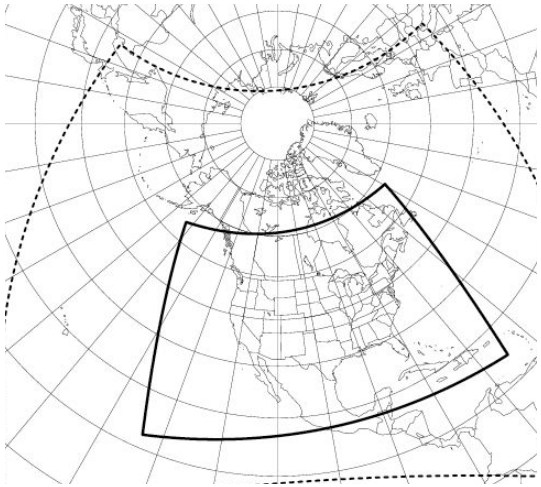
- ESMF_Grid - representation of a grid
- ESMF_State - objects that hold gridded data
- ESMF_State consists of ESMF_Field, ESMF_Bundle, ESMF_Array
- Data is exchanged between ESMF Gridded components using ESMF_States

# Key ESMF objects



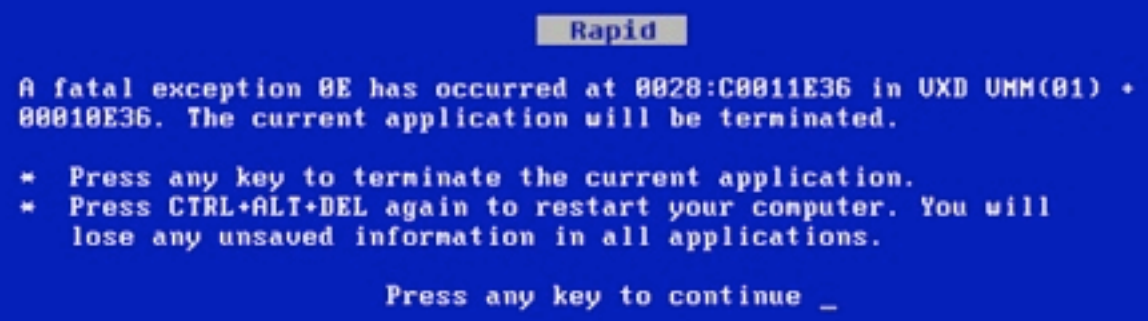
- ESMF_Grid - representation of a grid
- ESMF_State - objects that hold gridded data
- ESMF_State consists of ESMF_Field, ESMF_Bundle, ESMF_Array
- Data is exchanged between ESMF Gridded components using ESMF_States

# Key ESMF objects



- ESMF_Grid - representation of a grid
- ESMF_State - objects that hold gridded data
- ESMF_State consists of ESMF_Field, ESMF_Bundle, ESMF_Array
- Data is exchanged between ESMF Gridded components using ESMF_States

# Running LIS


Rapid
A fatal exception 0E has occurred at 0028:C0011E36 in VXD VMM(01) +
00010E36. The current application will be terminated.
* Press any key to terminate the current application.
* Press CTRL+ALT+DEL again to restart your computer. You will
lose any unsaved information in all applications.
Press any key to continue _

# Running LIS

Need to specify the lis.config file that contains the list of configurable options

Need parameter data (static, dynamic)

Need forcing data

How to get started?

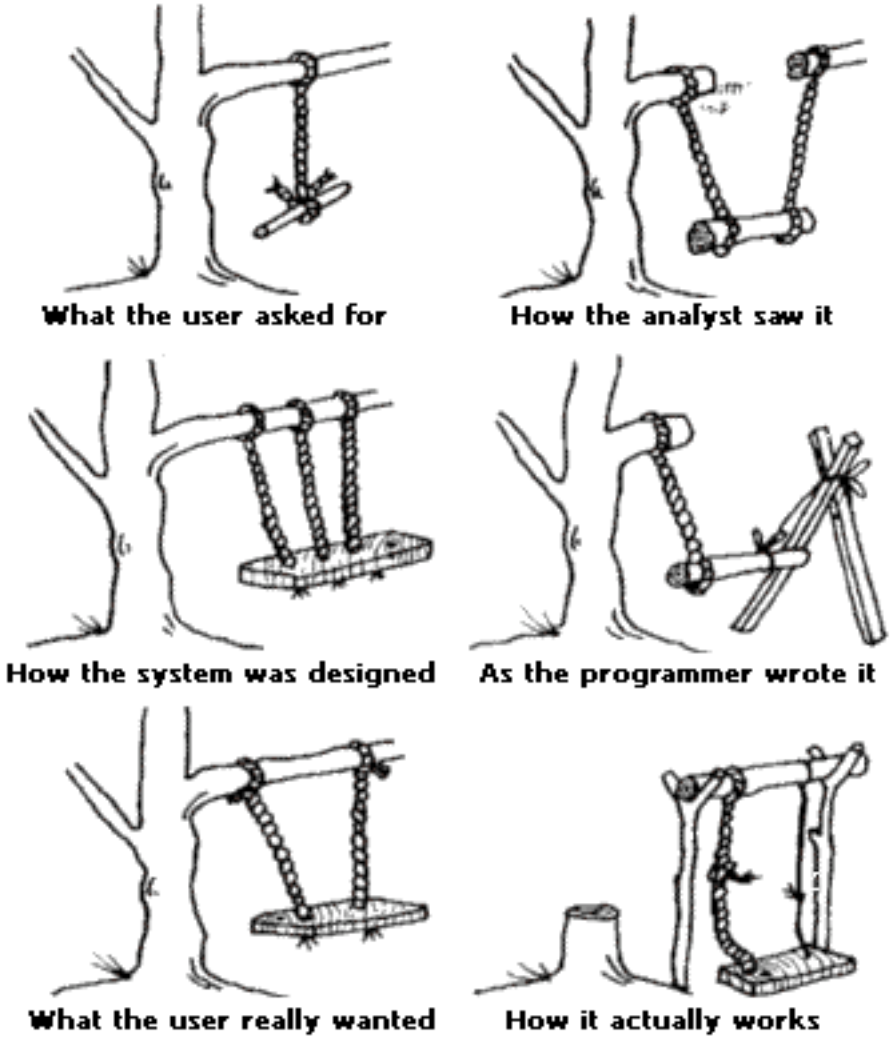
- Run the canned testcases: (http://lis.gsfc.nasa.gov/Source/testcases/)
- Find the lis.config file corresponding to each testcase under src/testcases

Generate parameter data at the desired resolution by using the LIS data processing programs

# Software Architecture

# Software Design

- Paradigm of "Problem Solving Environments" or "Expert Systems"
  - an integrated system provides the expert tools for complex domains
  - LIS is a PSE for hydrologic modeling applications
- LIS is designed as an object-oriented framework



# Object Oriented Programming

- Think Objects
- Modularity : Source code for an objects, written and maintained independent of the source code  for other objects
- Reusability: if the object already exists, you can use that object in your application
- Extensibility: Can be customized for new applications
- Inversion of Control - “Don’t call us, we’ll call you”

  Generic code controls execution of problem-specific code

States and Behavior

Gear
Speed
Pedal stance

change gear
apply brakes
speed up

# Object Oriented Programming

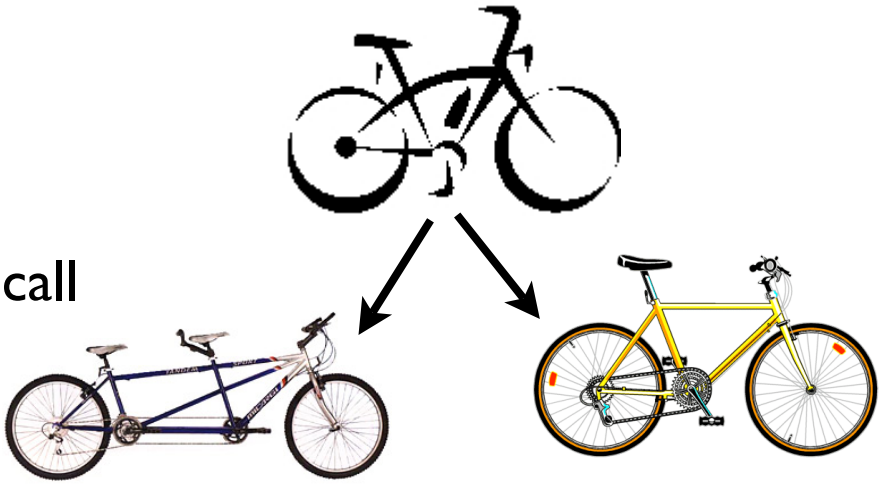
- Think Objects
- Modularity : Source code for an objects, written and maintained independent of the source code  for other objects
- Reusability: if the object already exists, you can use that object in your application
- Extensibility: Can be customized for new applications
- Inversion of Control - “Don’t call us, we’ll call you”

  Generic code controls execution of problem-specific code

# Object Oriented Programming

- Think Objects
- Modularity : Source code for an objects, written and maintained independent of the source code  for other objects
- Reusability: if the object already exists, you can use that object in your application
- Extensibility: Can be customized for new applications
- Inversion of Control - “Don’t call us, we’ll call you”

  Generic code controls execution of problem-specific code



# Object Oriented Programming

- Think Objects
- Modularity : Source code for an objects, written and maintained independent of the source code  for other objects
- Reusability: if the object already exists, you can use that object in your application
- Extensibility: Can be customized for new applications
- Inversion of Control - “Don’t call us, we’ll call you”

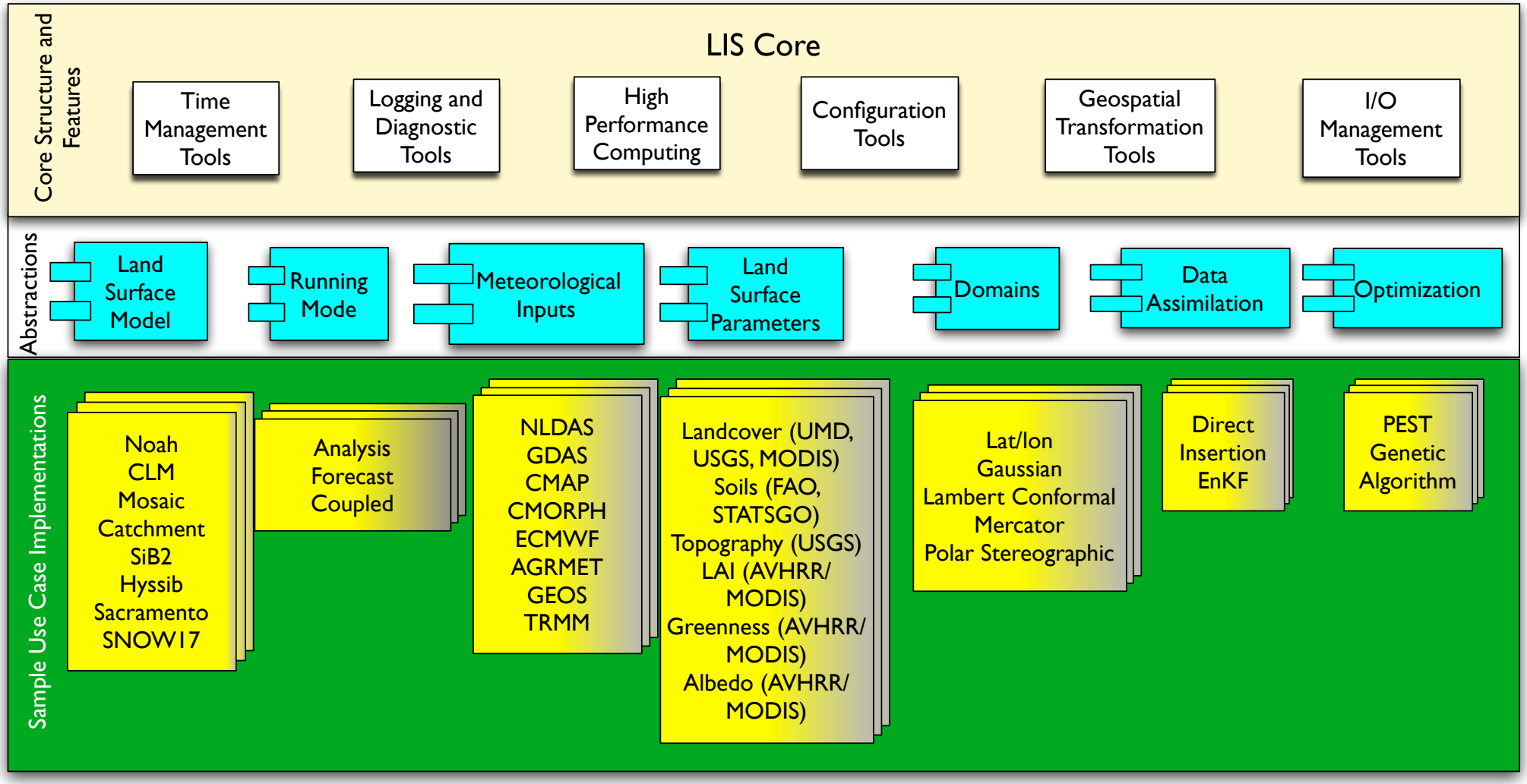
  Generic code controls execution of problem-specific code

# Object Oriented Programming

- Think Objects
- Modularity : Source code for an objects, written and maintained independent of the source code  for other objects
- Reusability: if the object already exists, you can use that object in your application
- Extensibility: Can be customized for new applications
- Inversion of Control - “Don’t call us, we’ll call you”

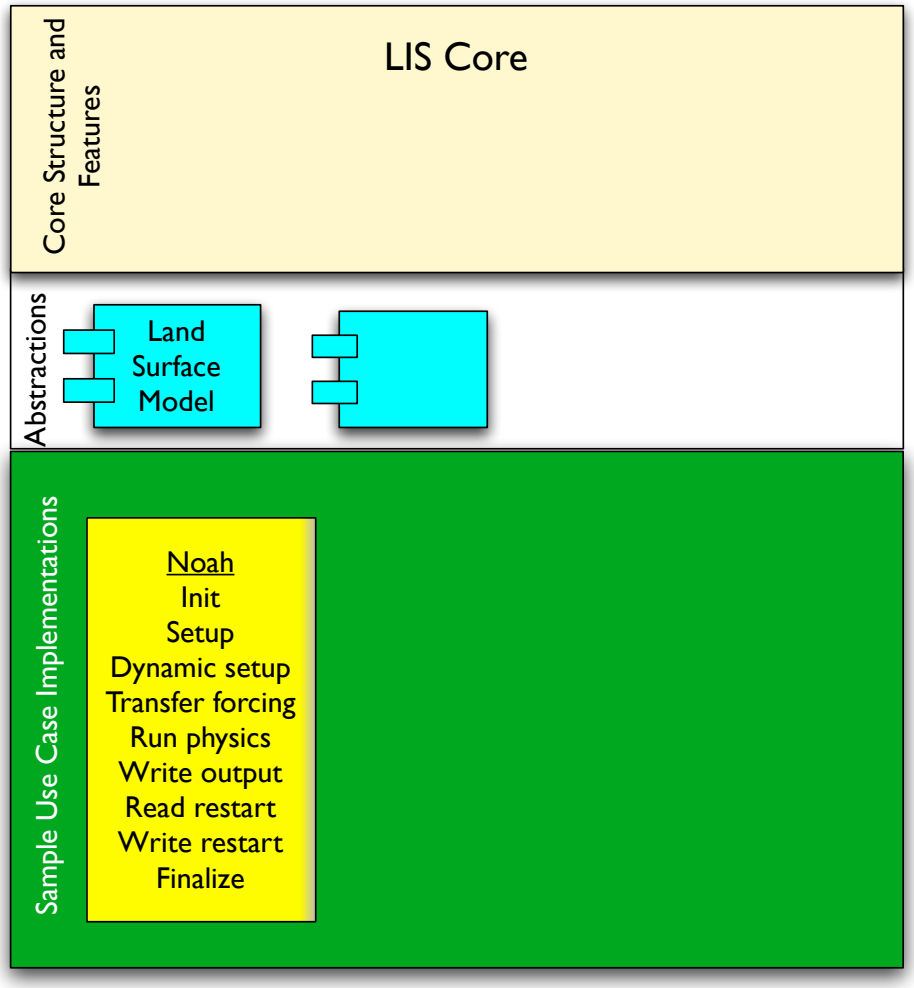
  Generic code controls execution of problem-specific code

# LIS Architecture



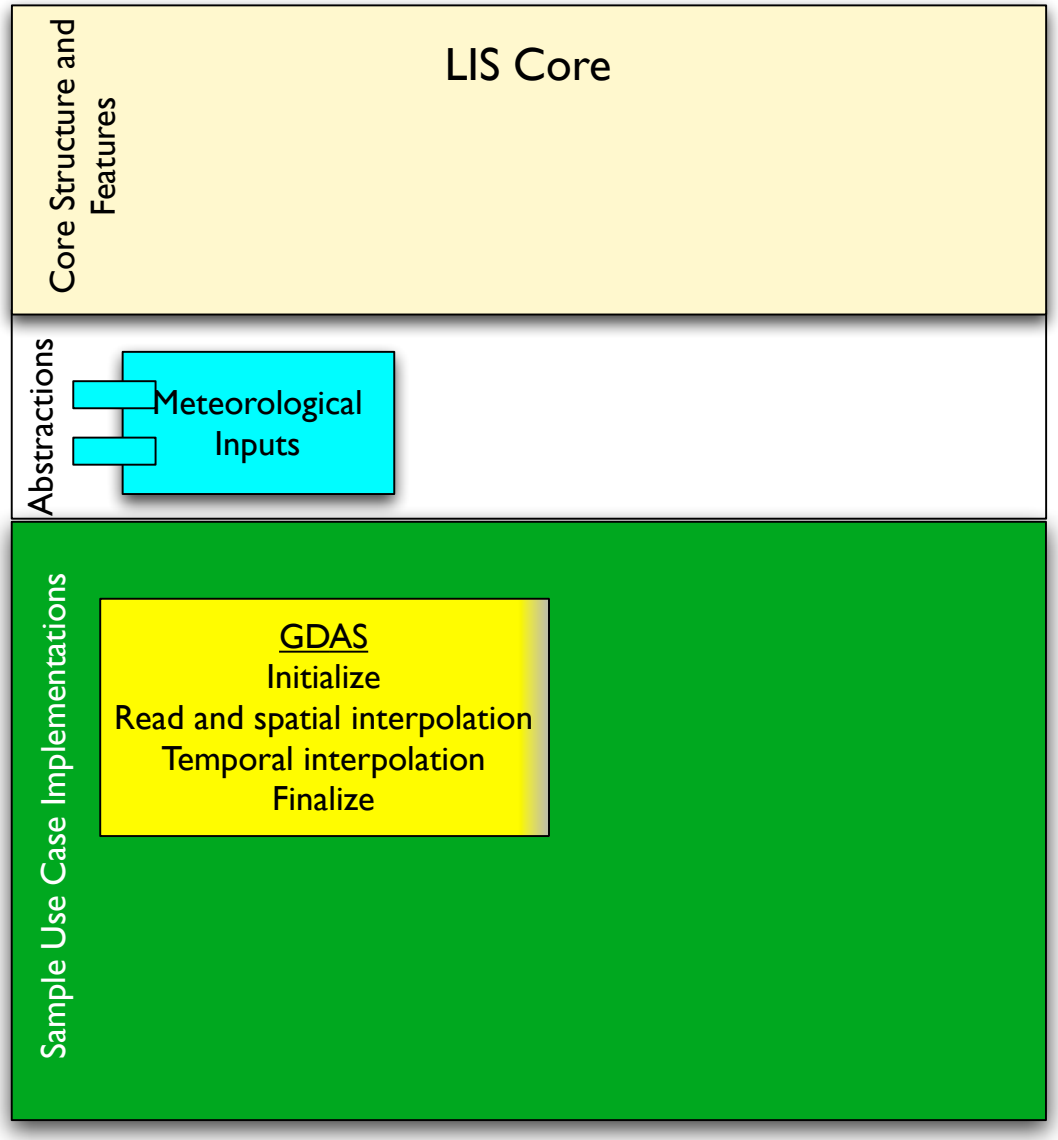
# Customizing LIS

# How do we add a new LSM?



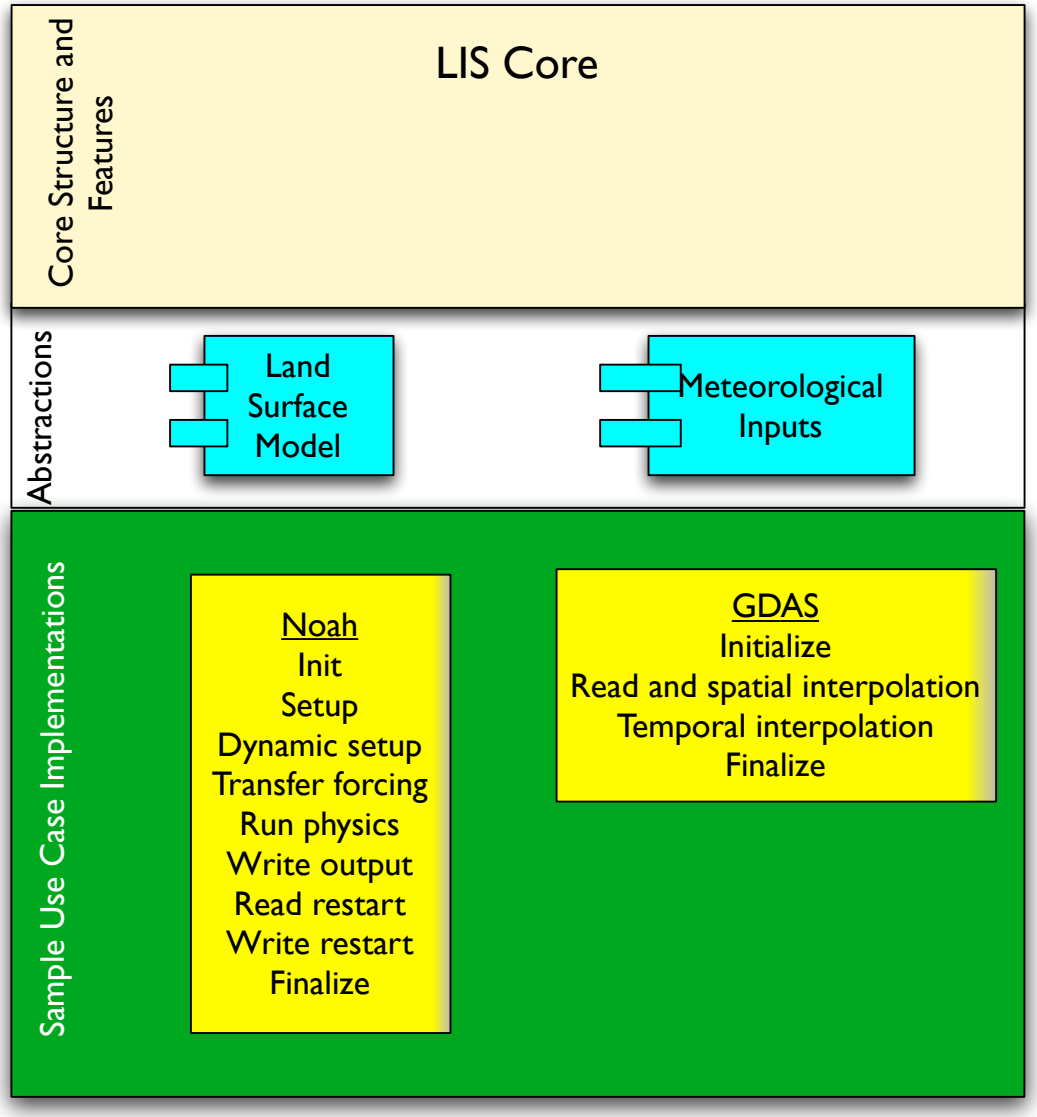

- Need to implement a set of interfaces related to the operation of a land surface model
- In LIS, these abstract implementations are known as "plugins"
- under src/plugins

# How do we add a new forcing scheme?




- Extend the abstract interfaces related to a forcing scheme

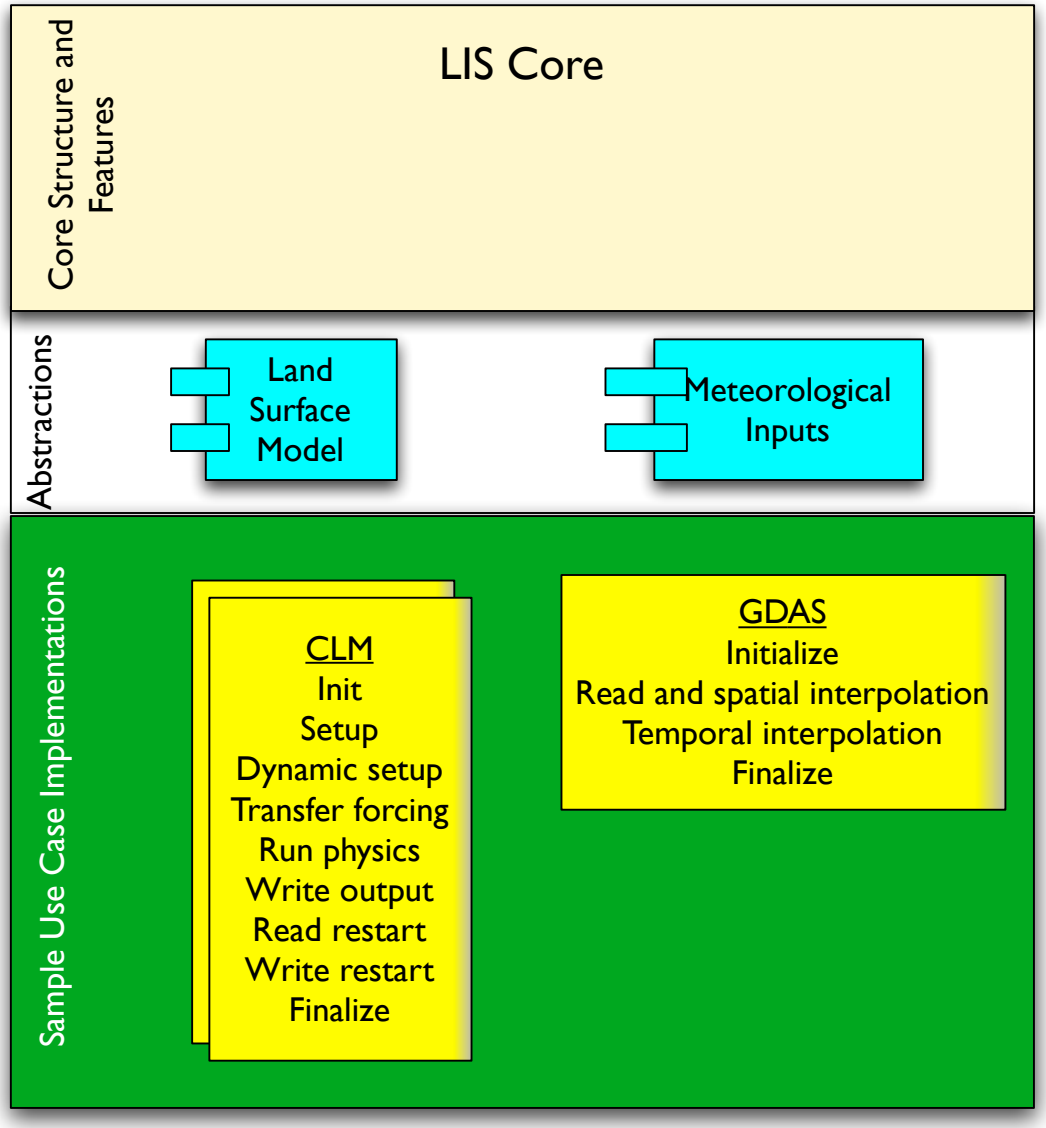
# Combining these components




LIS provides the "wirings" between the abstract implementations

Incorporating these components through plugins automatically ensure their integrated and interoperable use

# Combining these components

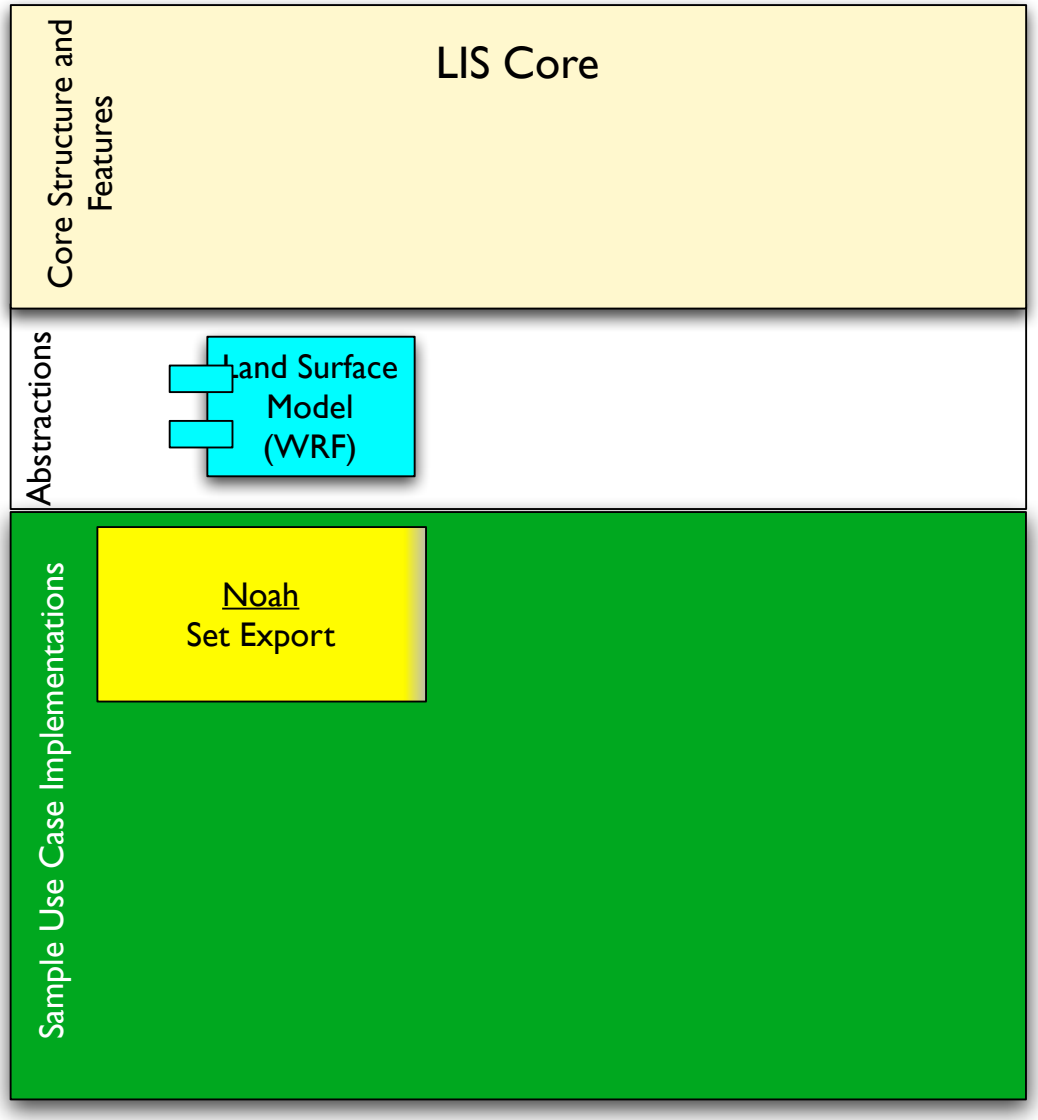



LIS provides the "wirings" between the abstract implementations

Incorporating these components through plugins automatically ensure their integrated and interoperable use

# Coupling to other earth system models (WRF)

# Enable coupling to WRF

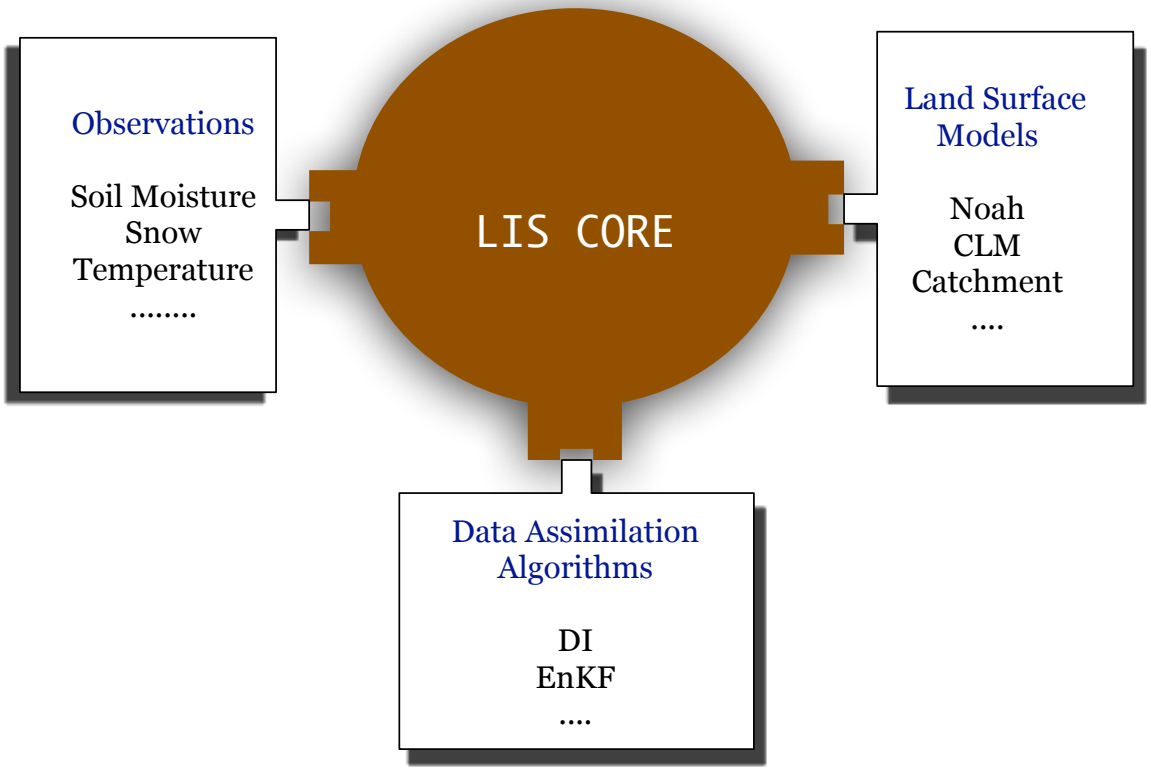



- ☑ A routine that specifies a set of export states to WRF need to be defined
- ☑ The import state to LIS is a standard list of forcing, common to all LSMs

**Kumar et al. (2007): An integrated high-resolution hydrometeorological modeling testbed using LIS and WRF, Environmental Modeling and Software, 23(2), 169-181.**

# Data Assimilation

# Abstractions related to Data Assimilation

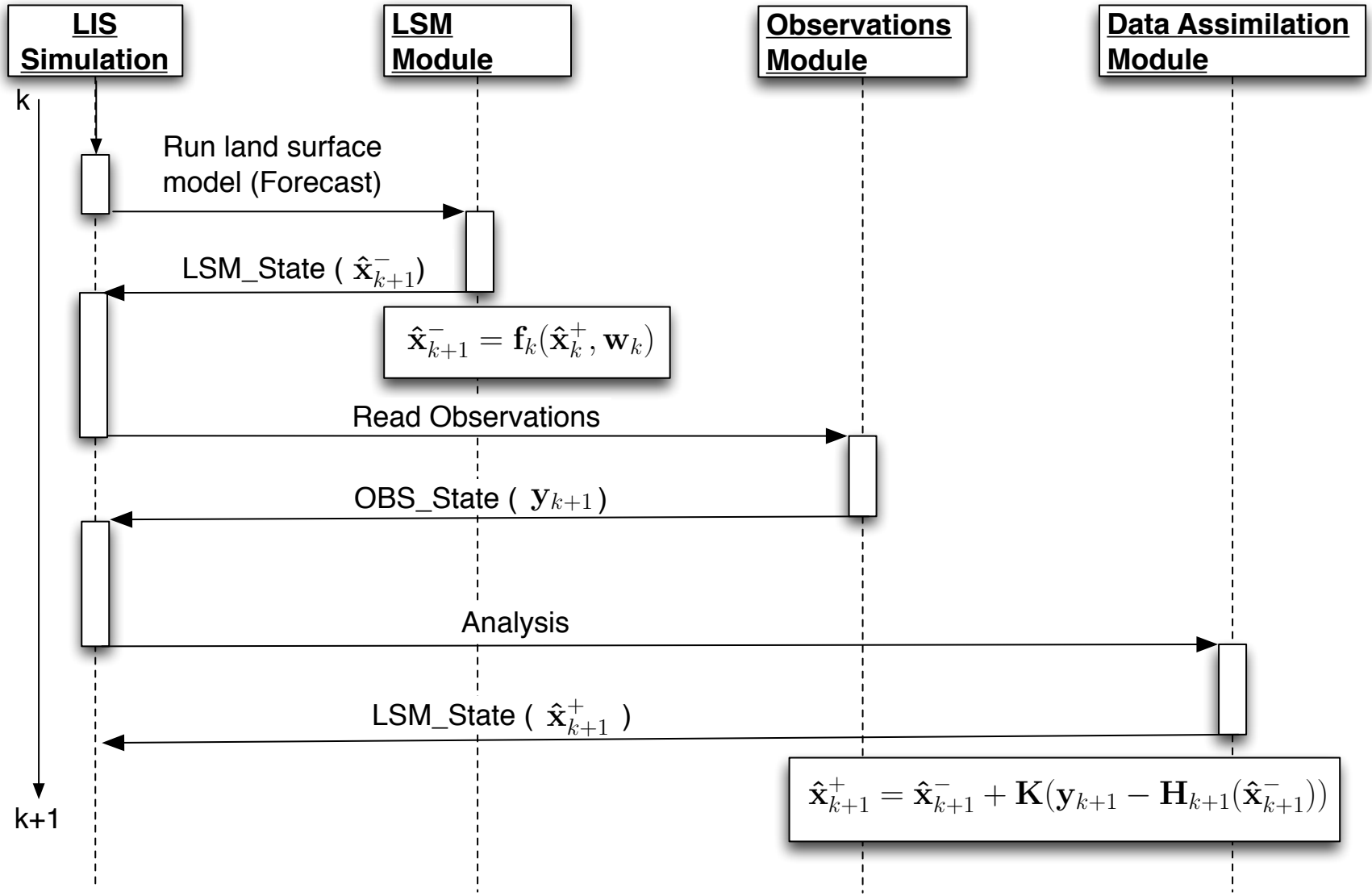



Goal:
Interoperability

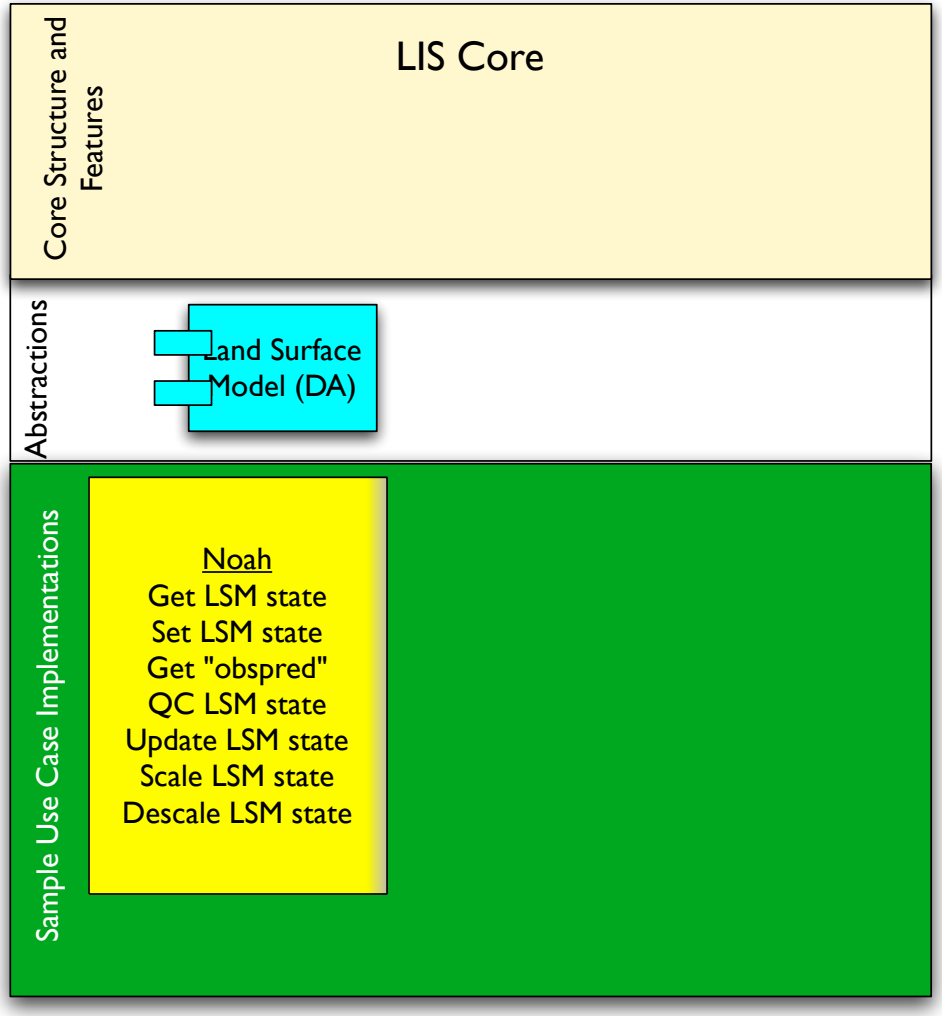
Once you define an observation plugin, it should work with an existing assimilation algorithm and LSM

**Kumar et al. (2008):A Land Surface Data Assimilation Framework using the Land Information System: Description and Application, Advances in Water Resources, doi:10.1016/j.advwatres.2008.01.013.**

# Sequence of component interactions

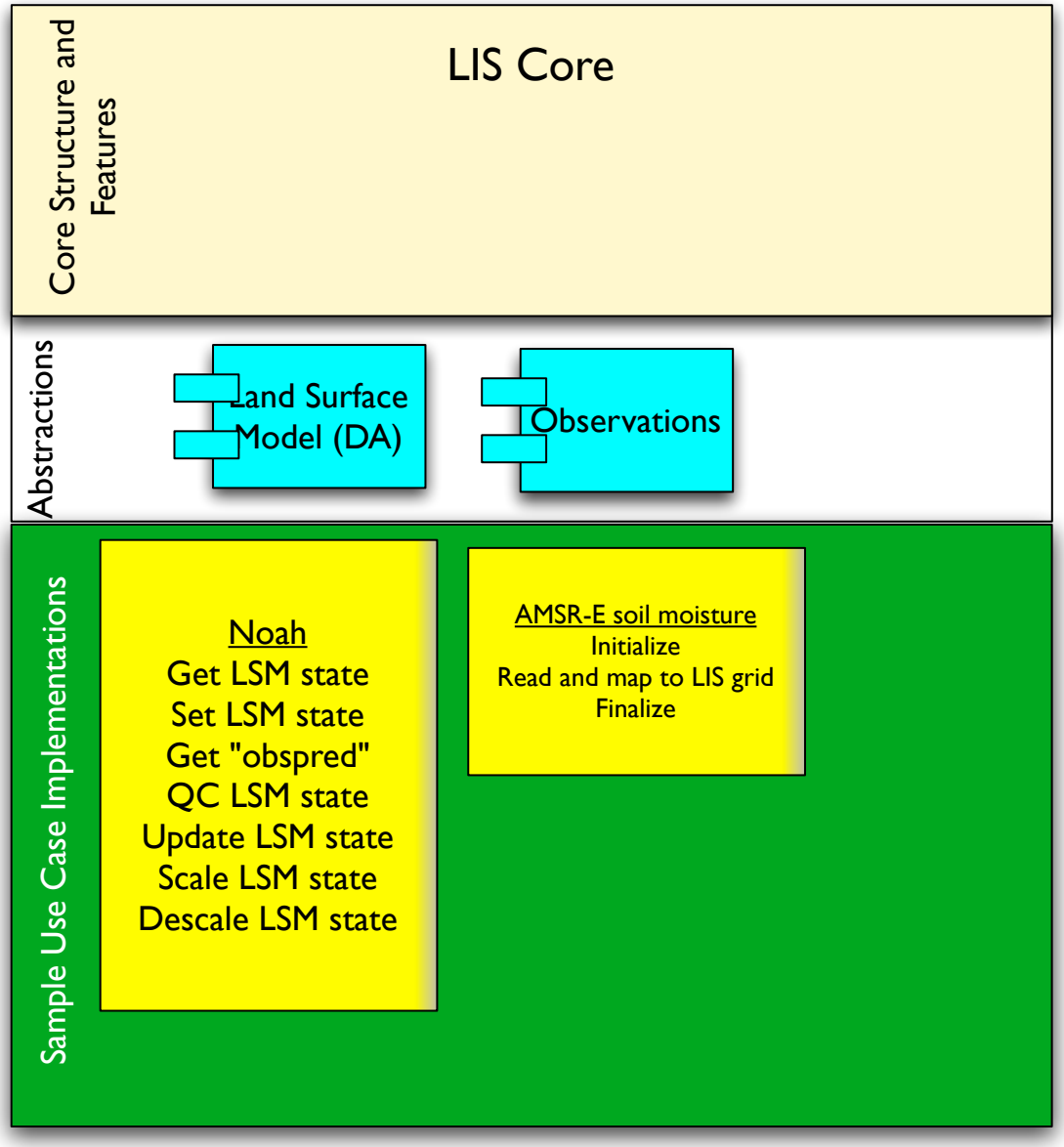


# How do we add a new DA instance?



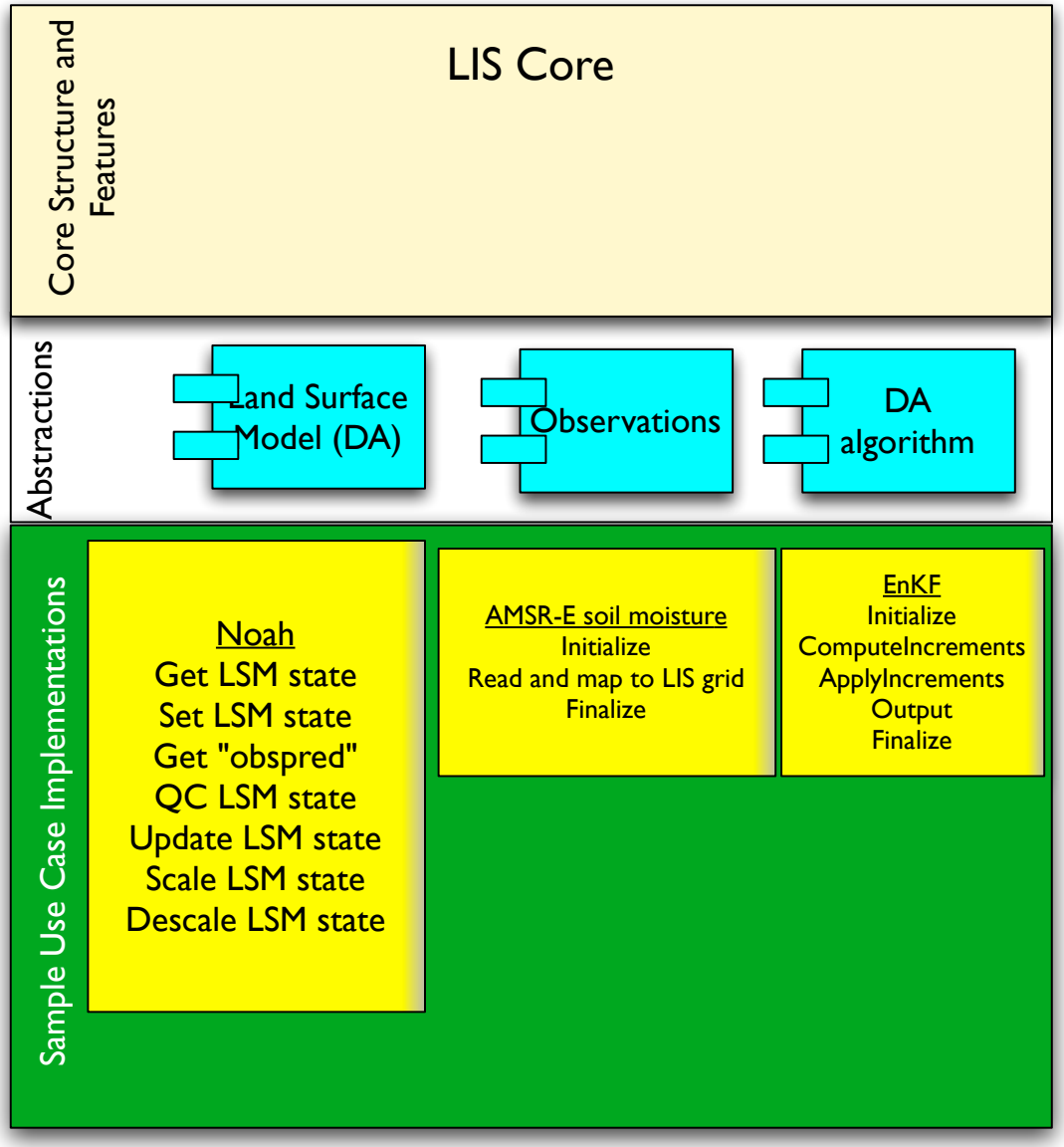

- Step 1
- ☐ Add DA related plugins for the LSM
  - ☑ Identify LSM prognostic variables
  - ☑ Define the "obspred" - what the model thinks the observation should be

# How do we add a new DA instance? (contd.)



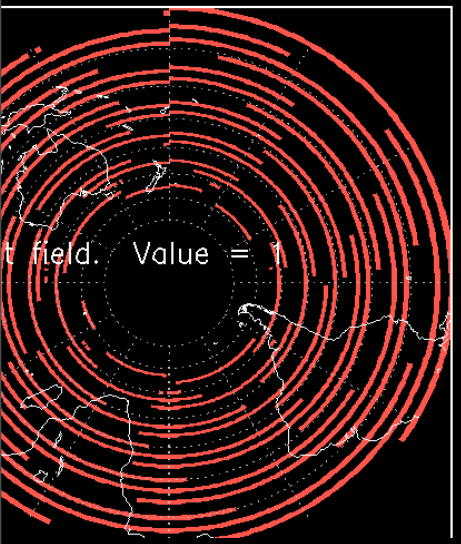

- Step2
- Add DA related plugins for observation source

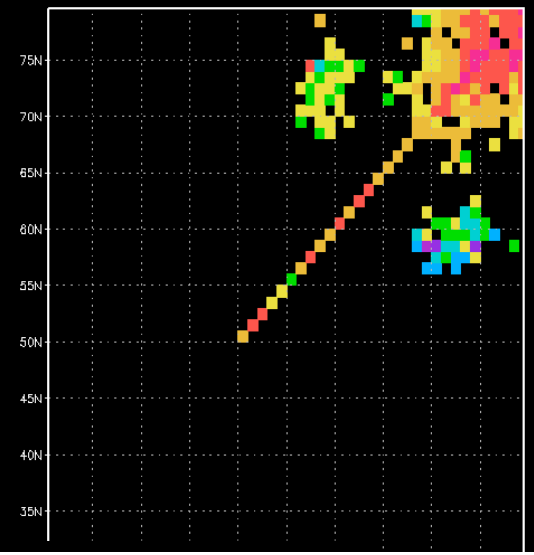
# How do we add a new DA instance? (contd.)

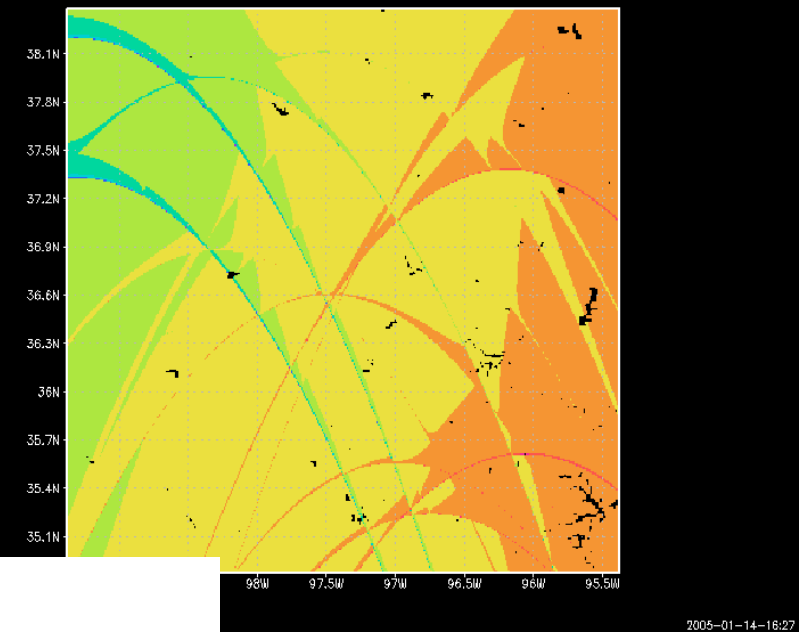



- Step3
- ☑ Add DA algorithm related plugins
- ☐ Users can simply reuse the existing implementation of EnKF and direct Insertion
- ☐ The wirings between these implementations are automatically done by virtue of the connections between the abstract implementations

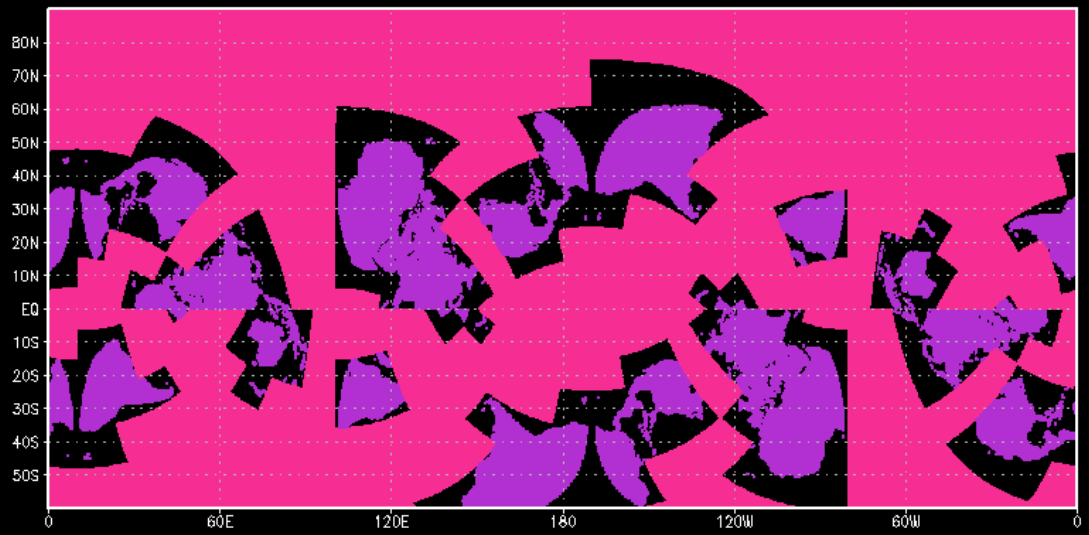
# LIS output

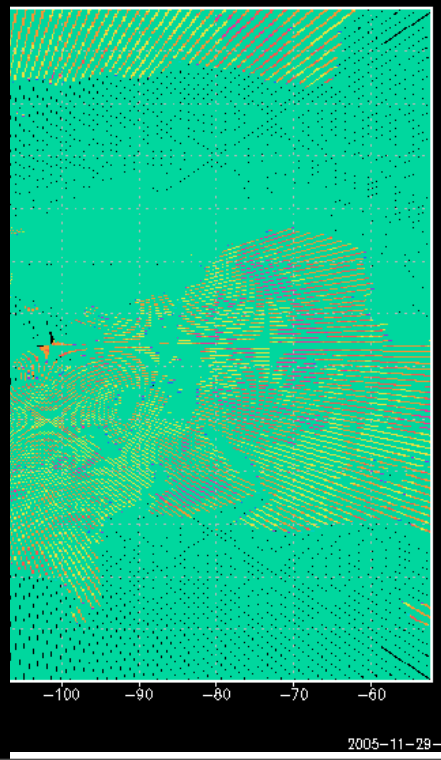

t field. Value = 1


75N
70N
65N
60N
55N
50N
45N
40N
35N


38.1N
37.8N
37.5N
37.2N
36.9N
36.6N
36.3N
36N
35.7N
35.4N
35.1N
98W
97.5W
97W
96.5W
96W
95.5W
2005-01-14-16:27

.5W 100W 99.5W 99W 98.5W 98W 97.5W 97W 96.5W 96W


80N
70N
60N
50N
40N
30N
20N
10N
EQ
10S
20S
30S
40S
50S
0
60E
120E
180
120W
60W
0


-100
-90
-80
-70
-60
2005-11-29-09:

# LIS output visualization

- Binary/Grib1/NETCDF output
- Use GrADS, IDL, etc.
- A few utilities: src/utils/
- src/utils/grads - program to generate a LIS control file
- src/utils/ensemble - program to generate a LIS ensemble restart file from a single member integration

Questions?