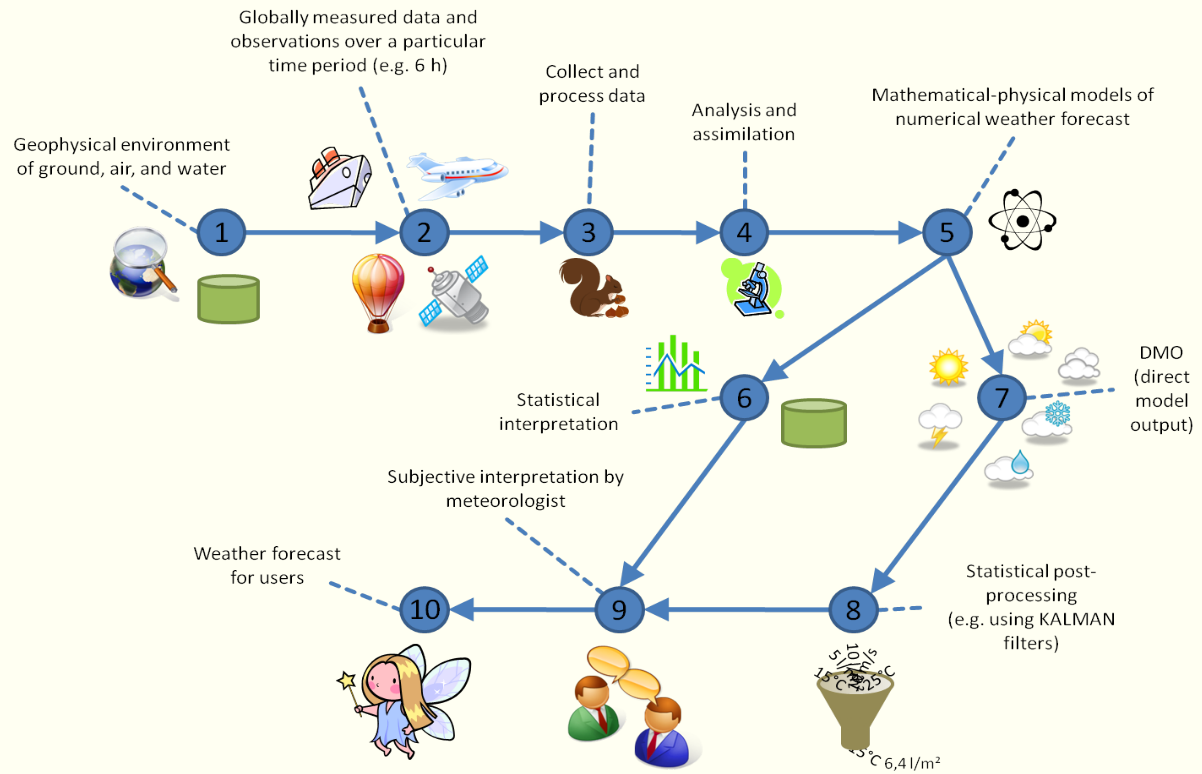
iRODS
Workflows & Beyond

Arcot (Raja) Rajasekar
UNC-Chapel Hill

Workflows

- orchestrated and repeatable pattern of activity (actions) that provide services, or process information.
  - Service-oriented workflows (eg. web services)
  - Data-oriented workflows (process information streams)

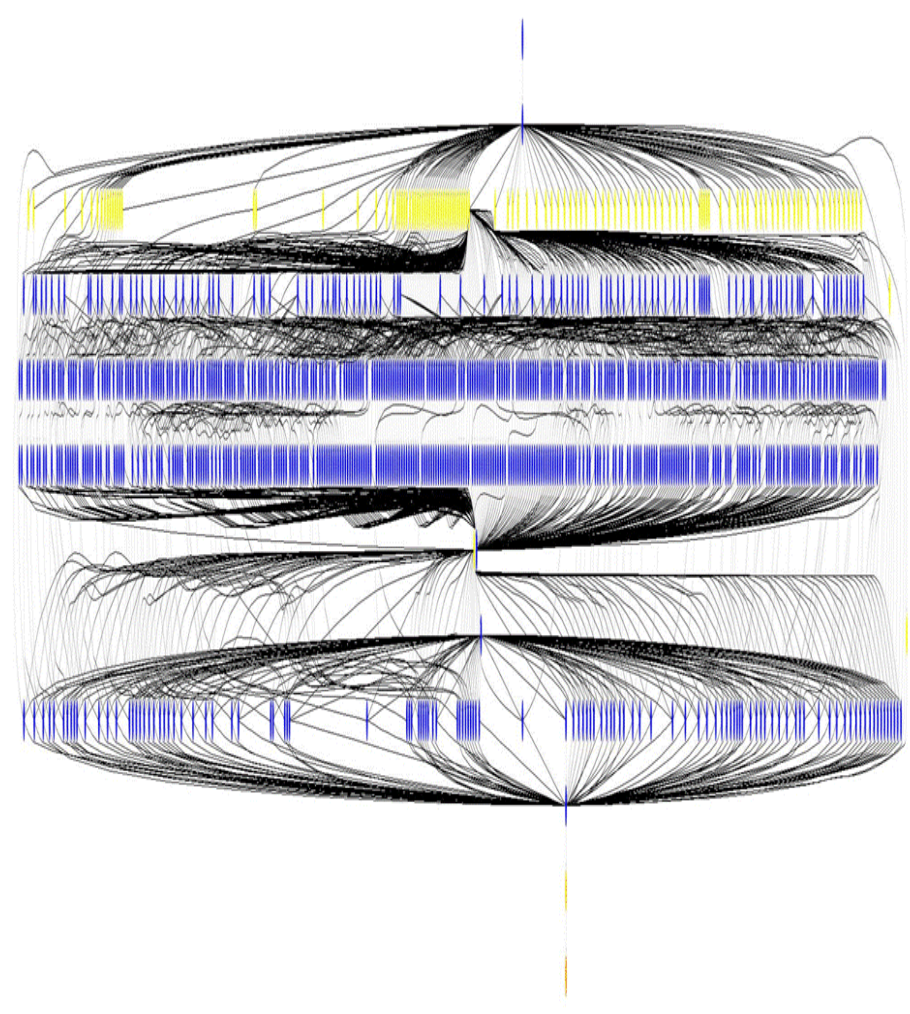


Data-Oriented Workflows

- **Workflows in the large - supercomputing**
  - Massive Computations on clusters/ clouds
  - Scheduling is important part of the process
  - Cluster-level data staging in and out essential
  - Example: Pegasus orchestration with

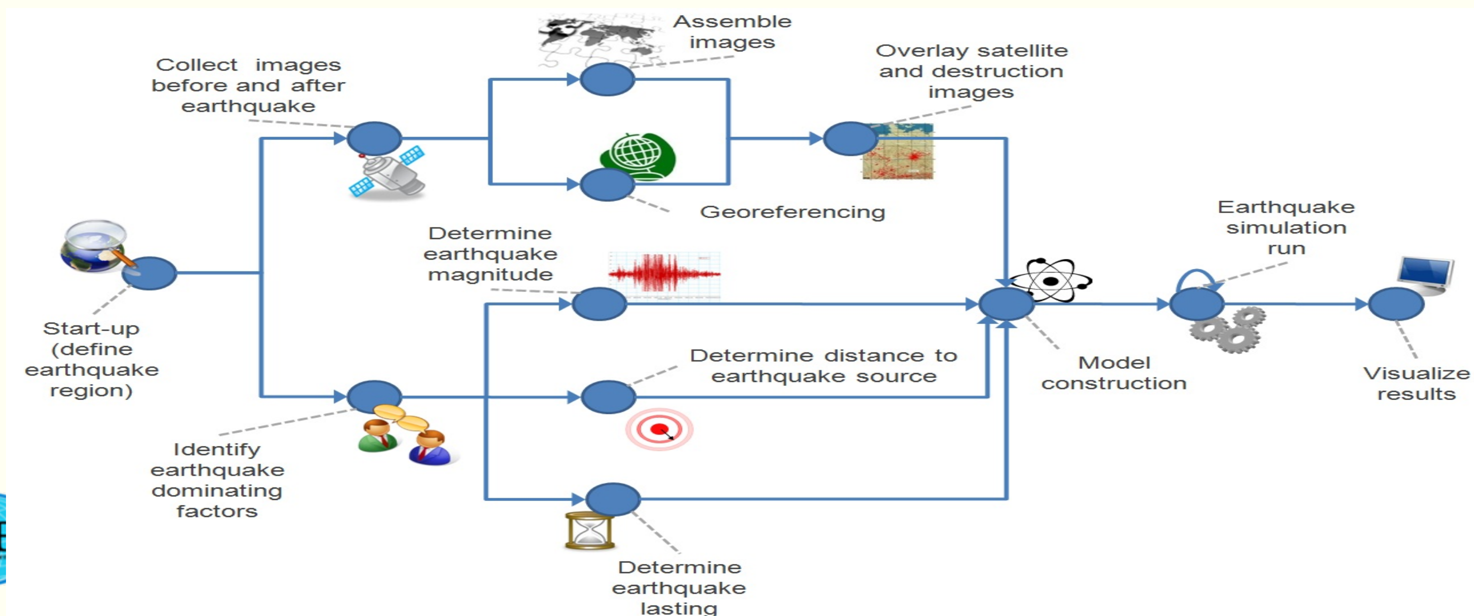
    DAGMan, Condor, SRM, Globus

    Parameter sweeps, simulations, etc.
    - Mostly linear (or tree) and few executables
  - Examples: Galactic Modeling, Earthquake Simulation



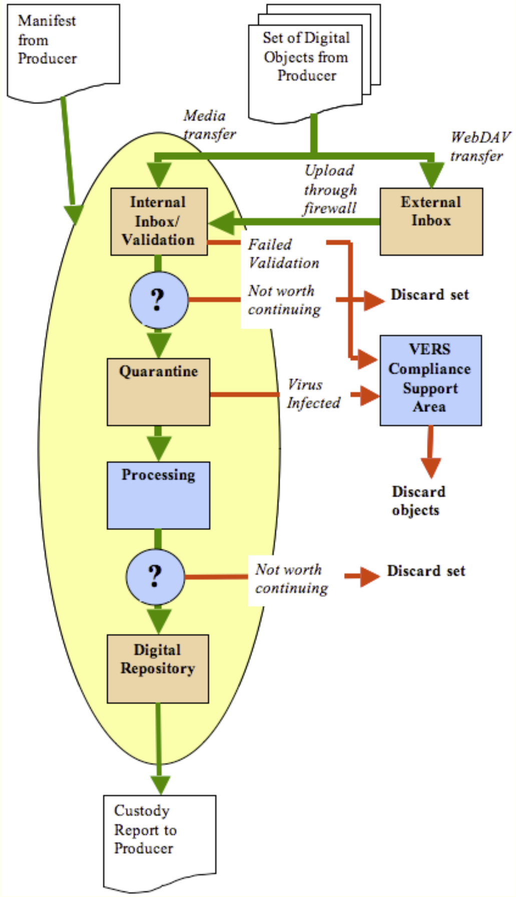
Data-Oriented Workflows

- **Workflows in the small - personal**
  - Small Computations on data servers
  - Scheduling & staging are not important
  - Data with High Variety & Veracity (2Vs of Big Data)
    - Multiple, distributed datasets
    - Data Preparation is essential
    - Provenance Capture is important
  - iRODS orchestration with Workflow Structured Object (WSO)
    - Coupled Models, Data Fusion, etc.
    - Reuse and Repurposing is key
  - Examples: Hydrologic Models, Biology Models



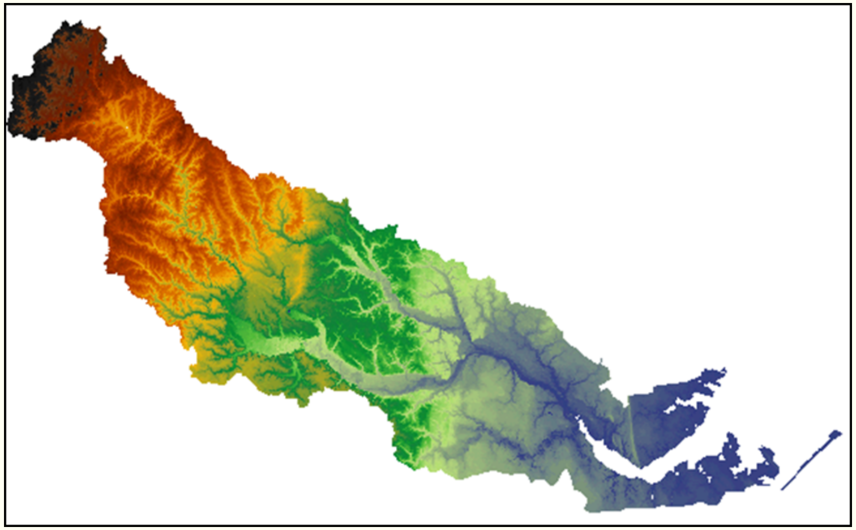
Workflow in the Small: VIC Model

- Variable Infiltration Capacity (VIC)
  - Macroscale Hydrologic Model
- VIC solves full water and energy balances,
- Developed by Xu Liang at the University of Washington and continued to be improved under the direction of Dennis P. Lettenmaier
- Has been applied to many watersheds including the Columbia River, the Ohio River, the Arkansas-Red Rivers, and the Upper Mississippi Rivers, as well as being applied globally.
- Adapted into iRODS by DFC partner Prof. Jon Goodall (UVa) for developing Watershed Models



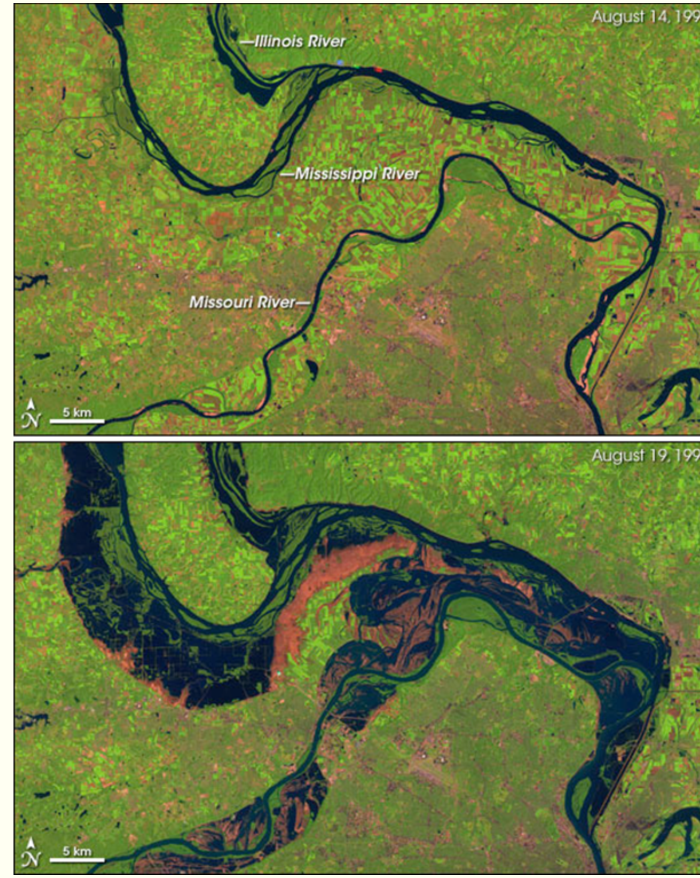
Datasets for VIC

- Meteorological Data Files
  - Precipitation, Wind, Temperature (Min, Max)
    - Precip & Temp Data obtained from
      - University of Washington (up to 1997)
      - NOAA/NCDC (National Climatic Data Center) (1998-current)
    - Wind Data (**NCEP/NCAR** reanalysis data) from
      - NOAA/ESRL (Earth System Research Lab)
- Soil & Vegetation Parameter Files
  - Both from NASA/LDAS (Land Data Assimilation System)
- Basin Shape Files
  - From USGS (USGSHYDRO1k shape file)



Preparation of Met Data

- **Precipitation & Temperature:**
  - Historical (Processed) Datasets (upto 12/1997)
    - Web access from U. of Washington
    - Access data for each State:
    - Daily data (*.dly) (compress after access) and station information (*.sta)
    - Example: NC: <ncprecip.dly.gz> and <ncprecip.sta>
  - Newer & Current Datasets (from 1/1998 till date) from NCDC
    - Raw file – accessible through web form Daily Surface Data by Country an State
    - Request Precip and Temp (Min,Max) data : all data in *.txt
      - email comes in giving links for ftp access
      - Four files accessed (data, station info, inventory, documentation)
      - Only data is needed as station is same as Historical data

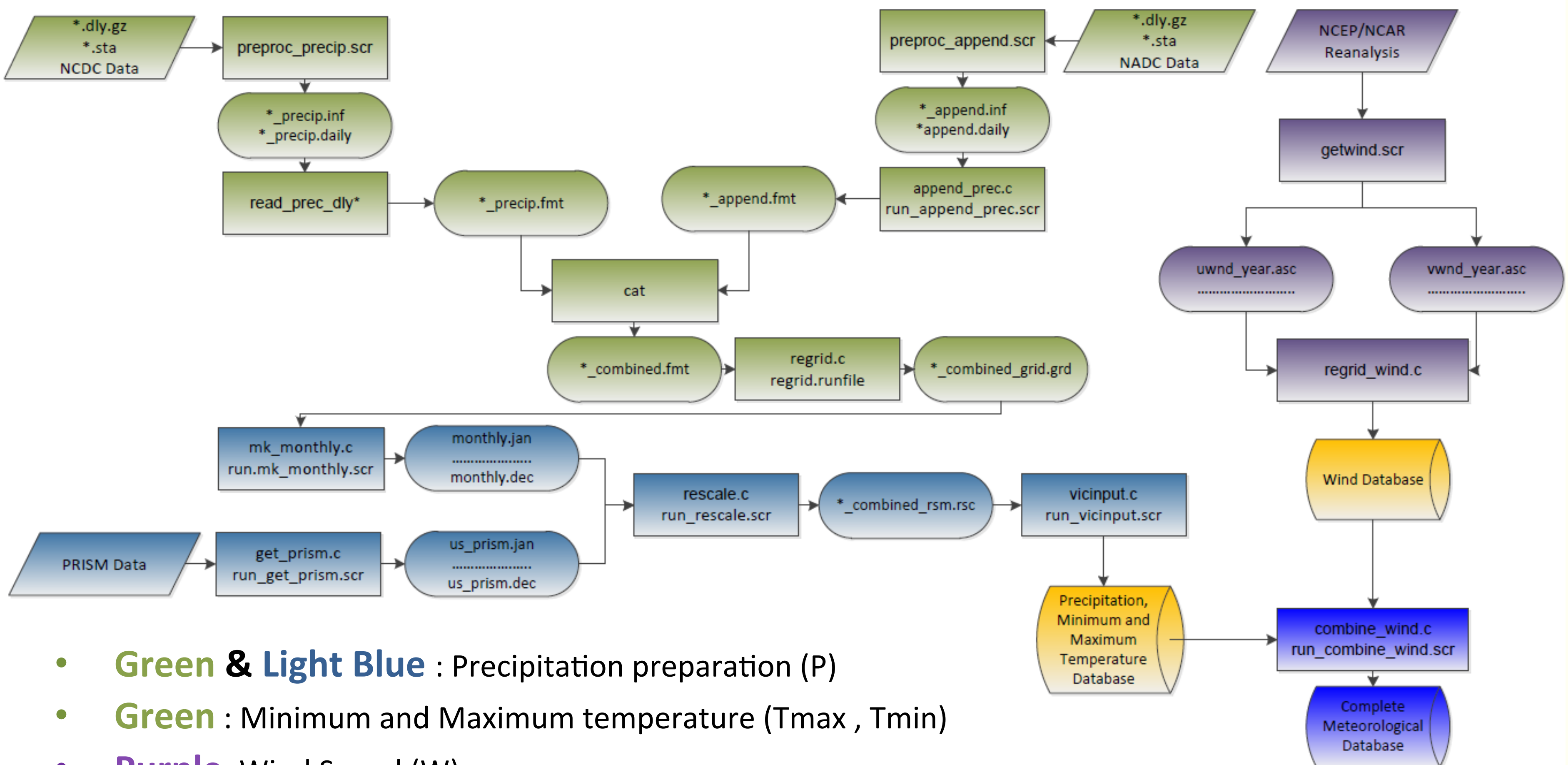


Pre-Processing Precip Files

- **GRID_200.TAR.GZ**
  - Contains Fortran and C files needed for processing
    - Example: NCAR-wind, NCDC-daily, regrid, vicinput, …
  - Pre-process Historical data
    - Script preproc_precip.scr from NCDC_daily is used to pre-process precipitation data
    - But, needs changes to the script by setting environment variables for output file locations
    - Run script with location of input files as parameter
    - Regrid precip file
      - Run Fortran executable read_prec_dly on output from last step
      - Output file preproc_prcp.fmt
  - Similar pre-process for Current Data from NCDC is also needed
  - Regrid and ReScale combined data (historical and current) to obtain monthly precipitation for each grid cell.
  - Monthly means of gridded precipitation output are scaled with monthly prism data for an entire basin which are combined with minimum and maximum temperature, and wind data before used as input for the VIC Model

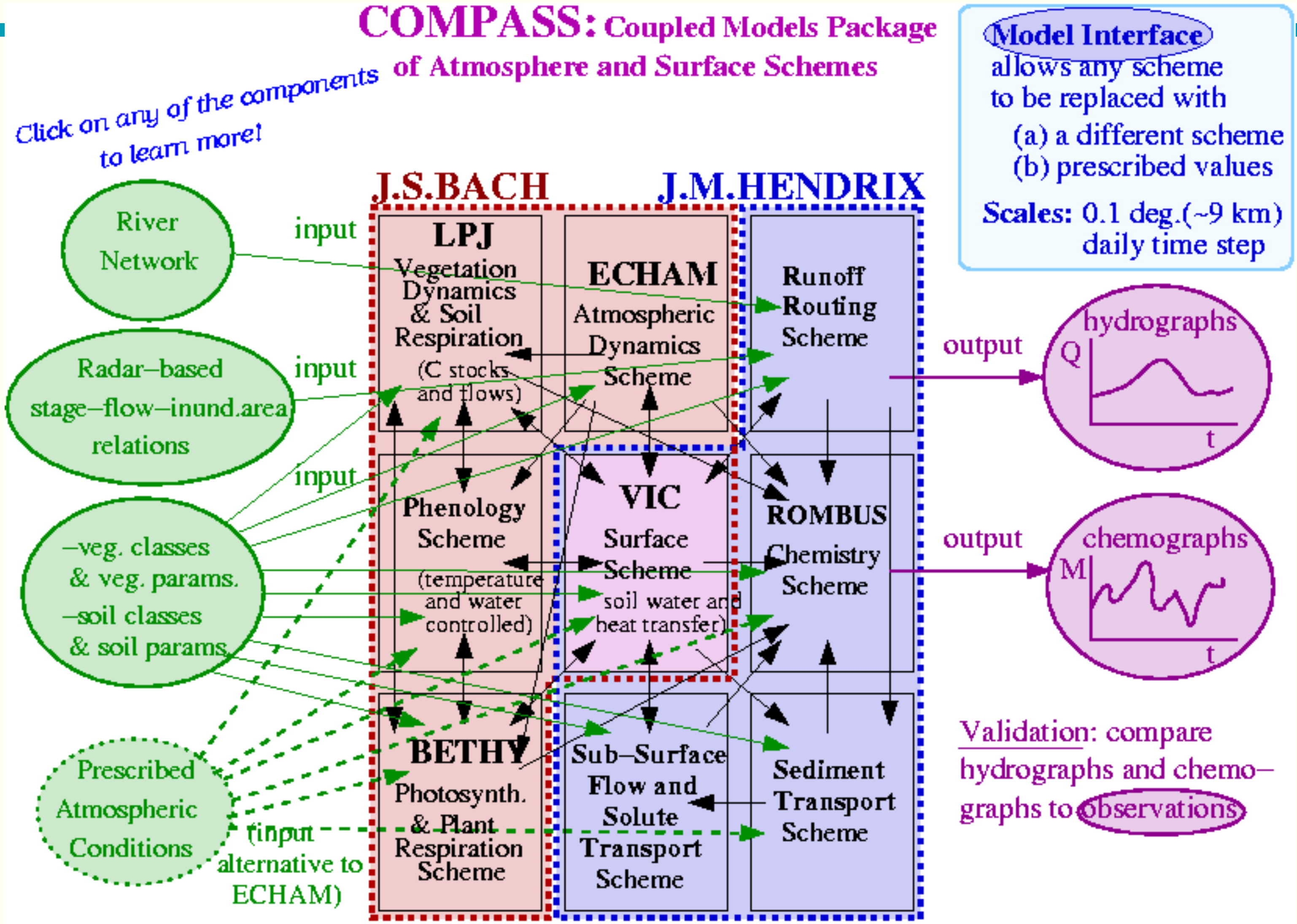
This is just pre-processing steps & just for Precipitation!

Schematic Diagram of Meteorological Forcing Data



- Green & Light Blue : Precipitation preparation (P)
- Green : Minimum and Maximum temperature (Tmax , Tmin)
- Purple: Wind Speed (W)
- Yellow: Combination of P, Tmax, Tmin, and separate W
- Blue : Combination of P, Tmax, Tmin, and W

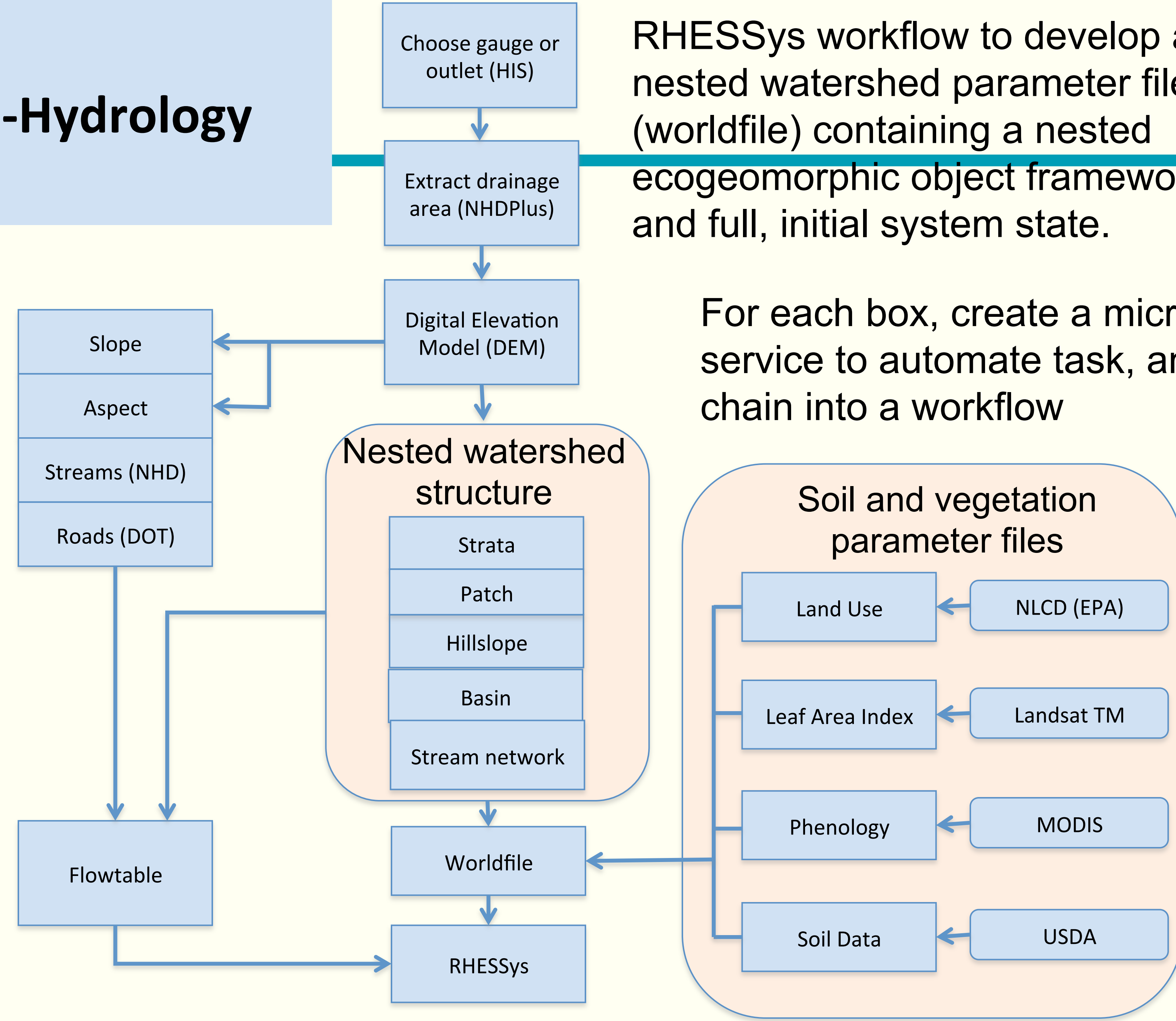
Larger Role of VIC



Eco-Hydrology

RHESSys workflow to develop a nested watershed parameter file (worldfile) containing a nested ecogeomorphic object framework, and full, initial system state.

For each box, create a micro-service to automate task, and chain into a workflow



iRODS Workflow

Capturing Provenance

eCWkflow.mss

/earthCube/eCWkflow

eCWkflow.run

eCWkflow2.run

eCWkflow.mpf

eCWkflow2.mpf

/earthCube/eCWkflow/eCWkflow.runDir0

Outfile

/earthCube/eCWkflow/eCWkflow2.runDir0

Newfile

Workflow file

Directory holding all input and output files associated with workflow file (mounted collection that is linked to the workflow file)

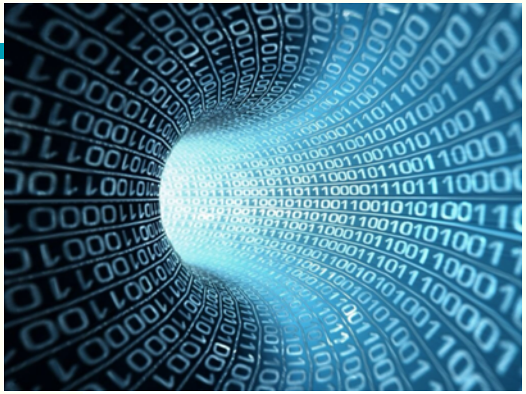
Automatically generated run file for Executing each input file

Input parameter file, lists parameters and input and output file names

Directory holding all output files generated from invocation of eCWkflow.run, the version number is incremented for each execution

Output files created for eCWKflow.mpf

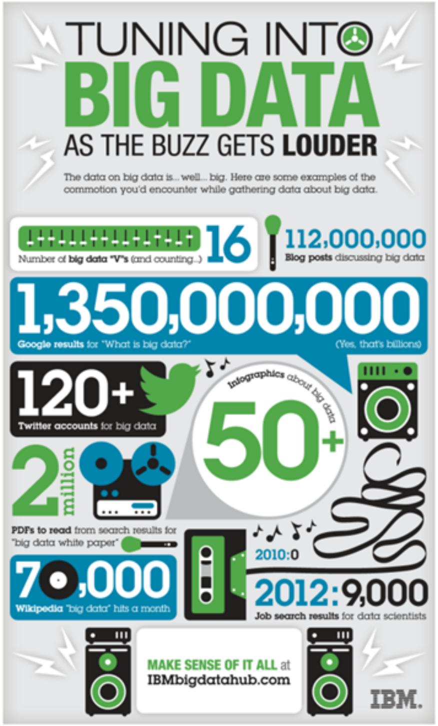
Paradigm Shift



iRODS is state of the art, BUT we are at a cusp

- Compute Intensive to
  Data Intensive
- Large Actions on Small Amounts of Data to
  Small Actions on Large Numbers of Data
- Move Data to Processing Site (Supercomputer)
  Move Process to Data Site (Map-Reduce)
- Function Chaining to
  Service Chaining
- Model-based Science to
  Data-based Science

(Data Mining, Knowledge Discovery)



What is the next leap in Workflows?

Data-driven Workflows (dataflow in the large)

On-demand (need based) Workflows

Loosely-coupled but strongly-typed Workflow Units

Multiple "vendors" - data shopping, job shopping

- Competition
- Service-level agreements (SLA)
- Different procedures
- Capability-driven, Capacity-driven
- Dynamically changing landscape

Self-orchestrated Workflows

How can we do this?

- Message-Oriented Architectures
- Active agents - everybody is an agent
  - Providing specific service computations
  - Subscribing to messages on a bus
  - No concept of physical end-points (from or to)
    - No hard-coding of service end-points
  - Logical message “headers” - service ontology/authority
  - Conversations leading to job/data shopping & SLAs
    - Trust and Value important   (aka yelp)
- Many agents can enter and leave the “marketplace”

D·I·C·E

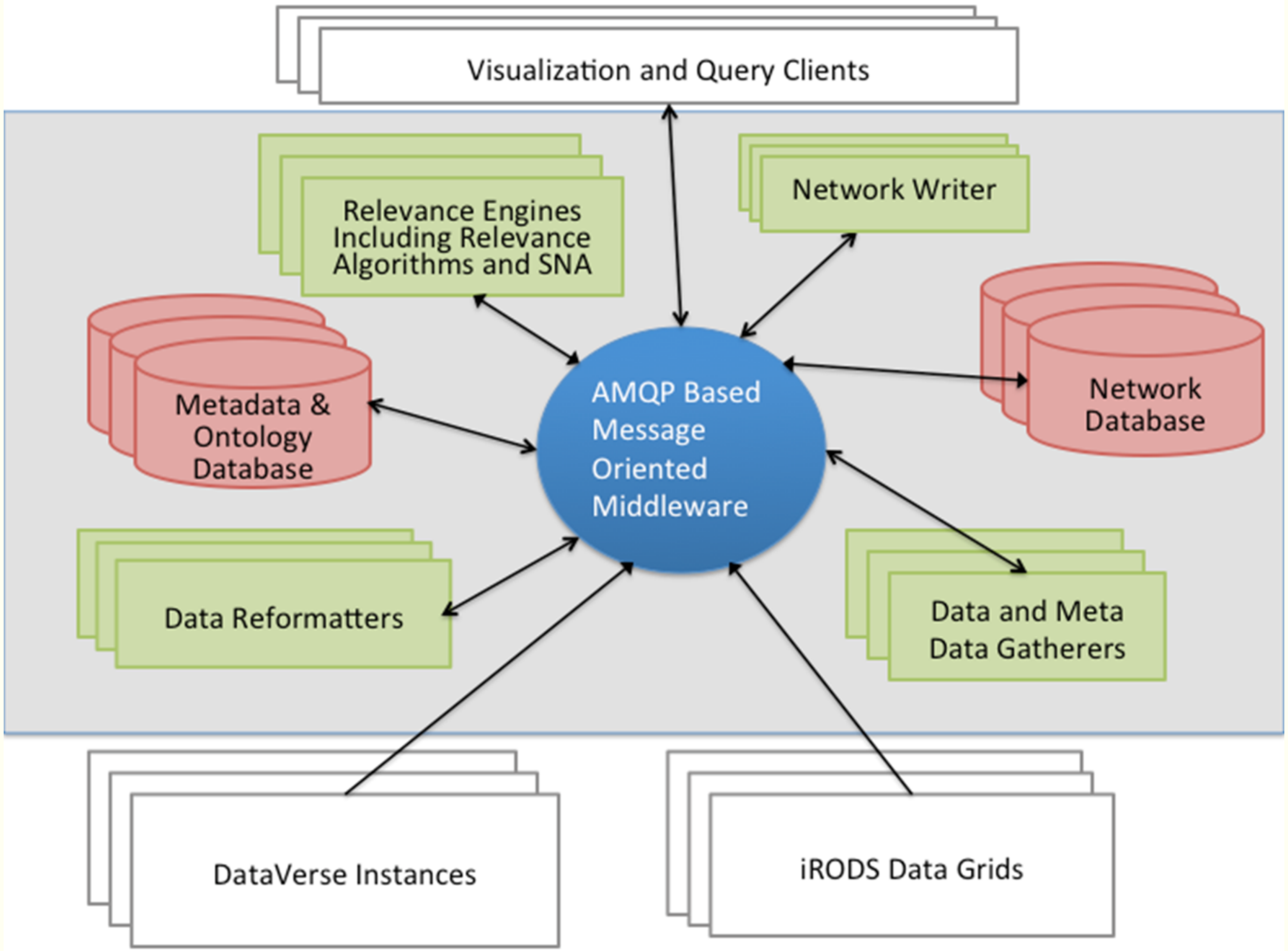
Data Intensive Cyber Environments

i·R·O·D·S

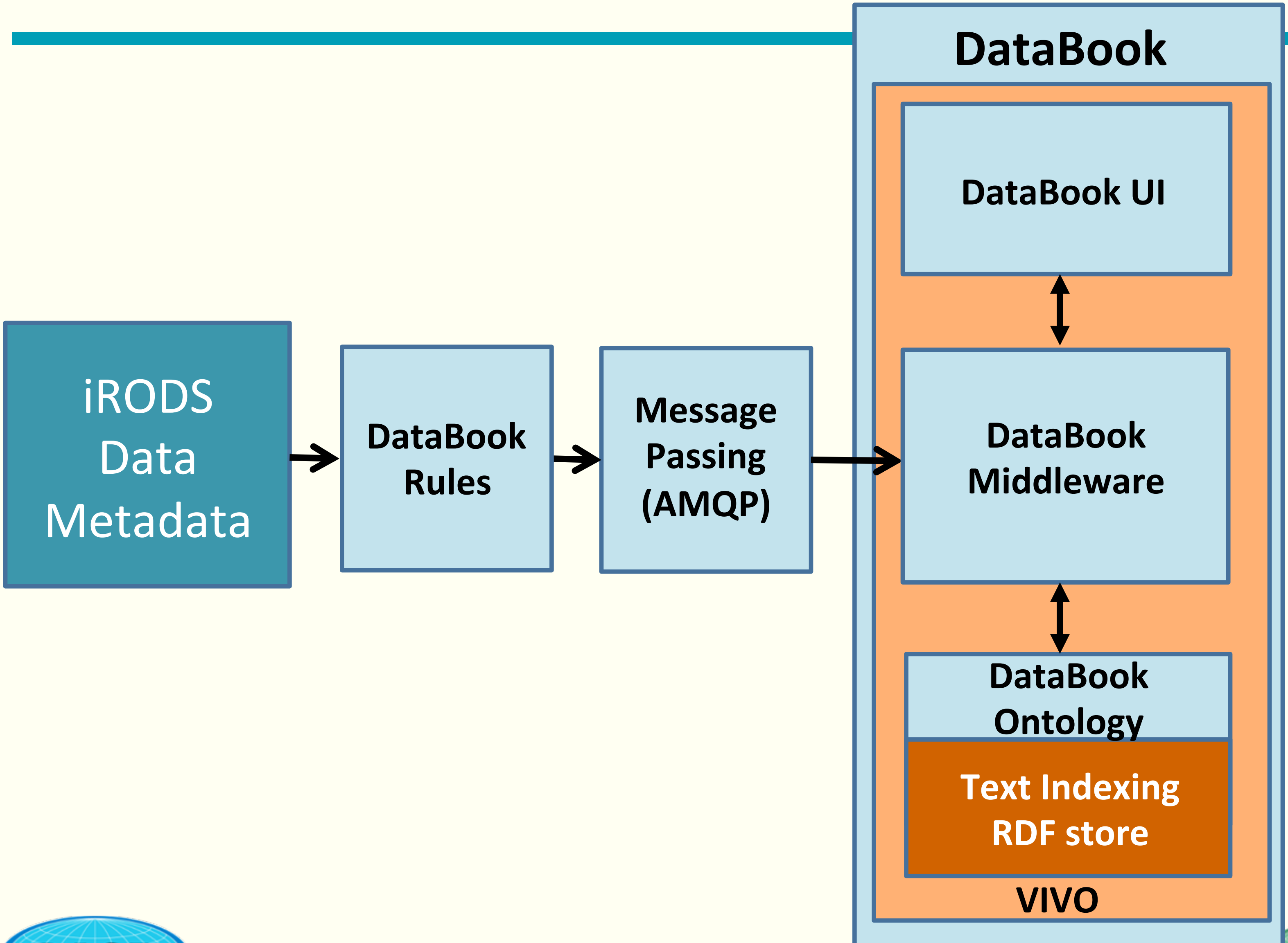
An Example System

- NSF Funded DataBridge Project
- Aim: Solve the Last Mile Problem
- Discover Interesting Relationships among data
  - Apply Socio-metric Network Analysis (SNA) to data
  - Link through Multi-dimensional vectors (clusters)
    - Similar to, but for data:
  - Explore Relationships between Data, Users, Resources, Methods, Workflows, …
  - Enable new discovery paradigms through different signature analyses
    - Message-oriented architecture fits right with this type of computation
    - As many signature agents can join and used for discovery
      - Its like Google allowing you to use your own "search & weightage algorithm"

Data Bridge Loosely-coupled Workflow Architecture



Another Example: DFC DataBook



Wrap up

- **Paradigm Shift**
  - Compute-intensive to data-intensive
  - Super-computing to Map-Reduce
  - Function-chaining to Service-chaining
  - Pre-rigged Orchestrated Workflows to Loosely-coupled Self-Orchestrated Workflows
  - Data is becoming the coin of the realm
    - Data drives the computation instead of being side-effect of computation
- **This is where we are going next …**

ILS - Tomorrow