

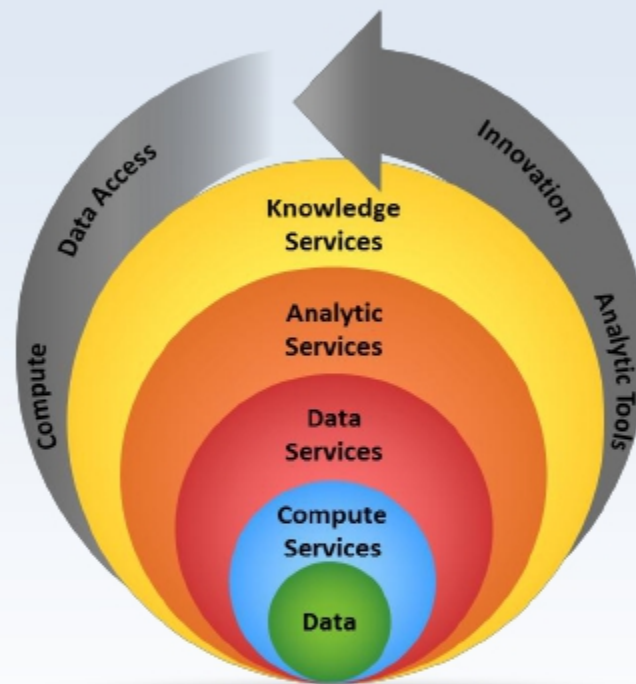
NASA Climate Model Data Services

CDS Mission:

Bringing together the tools, data storage and high-performance computing to for timely analysis over large-scale data sets, where the data resides, to ultimately produce societal benefits.

CDS Conceptual Service Layers:

- Compute Services
- Data Services
- Analytic Services
- Knowledge Services



*Serving the Earth Science
Community*

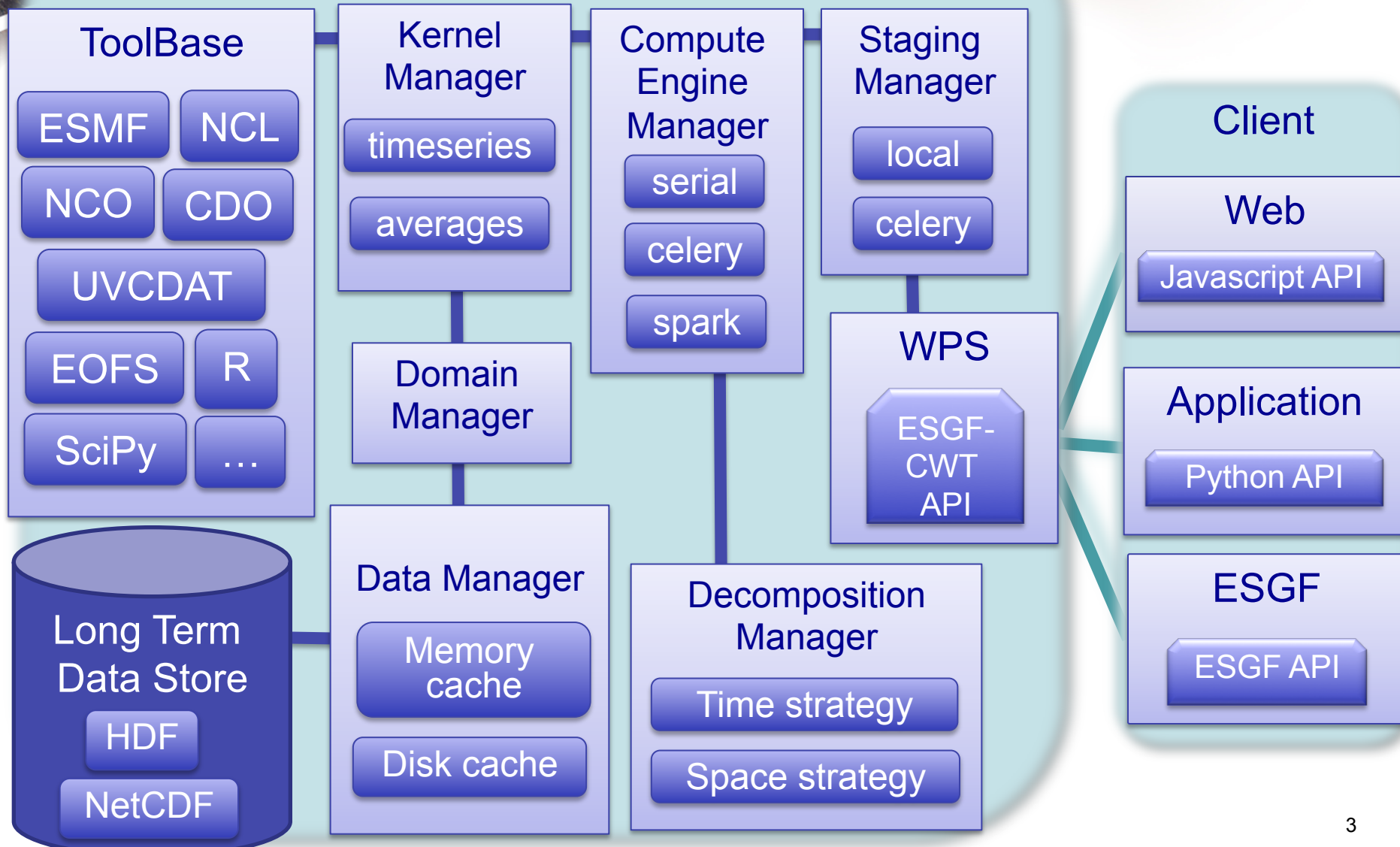


UVCDAT AaaS Requirements

- Access data in raw (NetCDF, HDF) format.
 - Avoid reformatting to support Hadoop, databases, etc.
 - Cache variables of interest in domain (xyzt) of interest.
- Support existing (python) climate data analysis operators.
 - All analysis operations developed in 100% python.
 - Parallelize data, not analysis packages.
- Support ESGF Compute Working Team API.
 - Utilize the OpenGIS Web Processing Service (WPS) Interface Standard.
- Modular design.
 - Easily integrate emerging technologies and support multiple approaches.
- Impose negligible overhead on interactive operations.
 - Light weight implementation using Flask.

Climate Data Analytic Services (CDAS)

NCCS-CDS AaaS Framework





Analysis Service Modules

Easily extendible compute components

- **Staging Methods:**
 - Local: Runs the operation locally (on the WPS server).
 - Celery: Uses Celery to run the operation on a remote server.
- **Compute Engines:**
 - Serial: Runs the computation locally on a single processor.
 - Celery: Uses Celery to distribute the computation over processors.
 - Spark: Uses Spark to distribute the computation over processors.
 - MPI: Uses MPI to distribute the computation over processors.
 - MultiProc: Uses python multiprocessing ** over cores.
- **Decomposition Strategies:**
 - Distributes the data over processors using the methods:
 - Time: Decomposes over time.
 - Space: Decomposes over space.

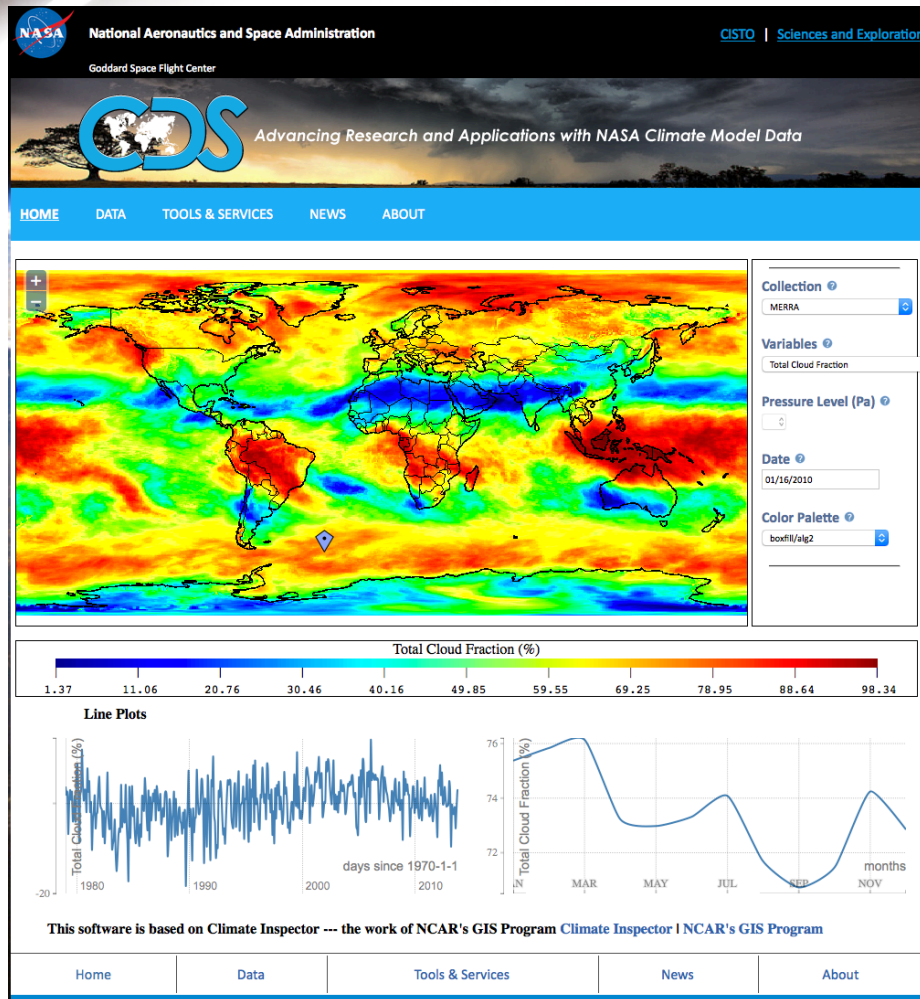


Analysis Service Modules

Easily extendible compute components

- **Compute Kernels:**
 - Encapsulated Computational Units exposed through WPS.
 - Executes a single processor's compute task.
 - Compatible with all CDAS compute engines and decompositions.
 - Implemented in python exploiting existing packages (e.g. UVCDAT)
- **Domain Manger:**
 - A domain consists of a:
 - List of variables-of-interest.
 - Spatio-temporal grid and region-of-interest.
 - Domain datasets are cached and reused whenever possible.
- **Data Manager:**
 - Reads domain data from long term storage.
 - Creates distributed memory-cached variables.
 - Transfers in-memory variables to/from disk cache when necessary.

CREATE-V CDAS Client



An interactive web application that expands GIS mapping capabilities for manual analysis of reanalysis data

Provides climate and non-climate scientists with ability to visualize reanalysis data, select variables, parameters and color maps

Permits decision makers to investigate climate changes around the globe, through time, inspect model trends, compare multiple reanalysis datasets, and variability



Timeline of Milestones

July - December 2015

- Deploy test CDAS WPS service for CREATE project (7/15).
- Implement CDAS caching manager (8/15).
- Implement celery-based parallelization (9/15).
- Harmonize WPS APIs: CDAS and ESGF-CWT (10/15).
- Coordinate efforts and exchange code with Kitware and LLNL (ongoing).
- Develop CDAS decomposition manager (11/15).
- Integrate additional cdutil functions into time compute kernel (12/15).
- Present results at ESGF-UVCDAT All Hands meeting (12/15)

January - June 2016

- Deploy operational CDAS WPS service for CREATE project (1/16)
- Implement spatial averaging compute kernel (2/16).
- Develop CDAS domain manager (3/16).
- Develop CDAS data manager with HDFS support (4/16).
- Integrate LLNL SLURM staging manager (5/16).