Future Generation Computer Systems 36 (2014) 400–417

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

# The Earth System Grid Federation: An open infrastructure for access to distributed geospatial data

Luca Cinquini [a,b,*], Daniel Crichton [a,b], Chris Mattmann [a,b], John Harney [c], Galen Shipman [c], Feiyi Wang [c], Rachana Ananthakrishnan [d,e], Neill Miller [d,e], Sebastian Denvil [f], Mark Morgan [f], Zed Pobre [g], Gavin M. Bell [h], Charles Doutriaux [h], Robert Drach [h], Dean Williams [h], Philip Kershaw [i,j], Stephen Pascoe [i,j], Estanislao Gonzalez [k,l], Sandro Fiore [m], Roland Schweitzer [n]

[a] *Jet Propulsion Laboratory (JPL), 4800 Oak Grove Drive, Pasadena, CA 91109, USA*
[b] *California Institute of Technology, 263 South Chester Avenue, Pasadena, CA 91106, USA*
[c] *Oak Ridge National Laboratory (ORNL), Oak Ridge (TN), USA*
[d] *University of Chicago, USA*
[e] *Argonne National Laboratory (ANL), Argonne (IL), USA*
[f] *Institut Pierre Simon Laplace (IPSL), Paris, France*
[g] *Goddard Space Flight Center (GSFC), Greenbelt (MD), USA*
[h] *Lawrence Livermore National Laboratory (LLNL), Livermore (CA), USA*
[i] *STFC Rutherford Appleton Laboratory, United Kingdom*
[j] *NCAS/BADC, Didcot, Oxfordshire, United Kingdom*
[k] *German Climate Computing Center (DKRZ), Hamburg, Germany*
[l] *Institute of Meteorology, Freie Universität Berlin, Germany*[1]
[m] *Euro-Mediterranean Center on Climate Change (CMCC), Lecce, Italy*
[n] *Pacific Marine Environmental Laboratory (PMEL), Seattle (WA), USA*

## HIGHLIGHTS

- ESGF is a global infrastructure to support climate change research.
- ESGF nodes around the world are serving tens of thousands of users.
- ESGF includes services for data discovery, access, analysis and visualization.
- ESGF is supporting operationally the CMIP5 global distributed archive (3PB).
- ESGF includes model output, observations, and reanalysis data.

## ARTICLE INFO

*Article history:*
Received 30 January 2013
Received in revised form
18 June 2013
Accepted 17 July 2013
Available online 17 September 2013

## ABSTRACT

The Earth System Grid Federation (ESGF) is a multi-agency, international collaboration that aims at developing the software infrastructure needed to facilitate and empower the study of climate change on a global scale. The ESGF's architecture employs a system of geographically distributed peer nodes, which are independently administered yet united by the adoption of common federation protocols and application programming interfaces (APIs). The cornerstones of its interoperability are the peer-to-peer messaging that is continuously exchanged among all nodes in the federation; a shared architecture and API for search

* Corresponding author at: Jet Propulsion Laboratory (JPL), 4800 Oak Grove Drive, Pasadena, CA 91109, USA.
*E-mail addresses:* luca.cinquini@jpl.nasa.gov (L. Cinquini), daniel.j.crichton@jpl.nasa.gov (D. Crichton), chris.a.mattmann@jpl.nasa.gov (C. Mattmann), harneyjf@ornl.gov (J. Harney), gshipman@ornl.gov (G. Shipman), fwang2@ornl.gov (F. Wang), ranantha@mcs.anl.gov (R. Ananthakrishnan), neillm@mcs.anl.gov (N. Miller), sebastien.denvil@ipsl.jussieu.fr (S. Denvil), momipsl@ipsl.jussieu.fr (M. Morgan), zed.pobre@nasa.gov (Z. Pobre), gavin@llnl.gov (G.M. Bell), doutriaux1@llnl.gov (C. Doutriaux), drach1@llnl.gov (R. Drach), williams13@llnl.gov (D. Williams), philip.kershaw@stfc.ac.uk (P. Kershaw), stephen.pascoe@stfc.ac.uk (S. Pascoe), estanislao.gonzalez@met.fu-berlin.de (E. Gonzalez), sandro.fiore@unisalento.it (S. Fiore), Roland.Schweitzer@noaa.gov (R. Schweitzer).

[1] Present address.

0167-739X/$ – see front matter 
http://dx.doi.org/10.1016/j.future.2013.07.002

*Keywords:*
Climate science
Federation
Search
Discovery
Peer-to-peer
CMIP5

and discovery; and a security infrastructure based on industry standards (OpenID, SSL, GSI and SAML). The ESGF software stack integrates custom components (for data publishing, searching, user interface, security and messaging), developed collaboratively by the team, with popular application engines (Tomcat, Solr) available from the open source community. The full ESGF infrastructure has now been adopted by multiple Earth science projects and allows access to petabytes of geophysical data, including the entire Fifth Coupled Model Intercomparison Project (CMIP5) output used by the Intergovernmental Panel on Climate Change (IPCC) Fifth Assessment Report (AR5) and a suite of satellite observations (obs4MIPs) and reanalysis data sets (ANA4MIPs). This paper presents ESGF as a successful example of integration of disparate open source technologies into a cohesive, wide functional system, and describes our experience in building and operating a distributed and federated infrastructure to serve the needs of the global climate science community.



## 1. Introduction

The study of climate change, including an evaluation of its impact on Earth's ecosystem and human society, is one of the most important scientific challenges of our time. Because the physical processes governing Earth's climate are extremely diverse and complex, this research involves: (1) sophisticated model simulations that generate unprecedented amounts of output data; and (2) the collection of observational data from multiple sources (such as remote sensors, in situ probes, and vertical profiles) on a global scale. These data sets are managed and stored at multiple geographic locations across the globe; yet, they need to be discovered, accessed, and analyzed as if they were stored in a single centralized archive.

For over a decade, the collaborative objective has been to establish the Earth System Grid Federation (ESGF) as a leader and visionary architect for all aspects of climate data discovery and knowledge integration. From its formative years (previously known as the Earth System Grid (ESG) [1]), ESGF has demonstrated a viable path forward into the future and attracted key national and international collaborators and sponsors recently reaching the critical mass required to undertake fundamental data-intensive and data-driven science problems. In the climate application arena, an increasing number of projects are consolidating on the ESGF peer-to-peer (P2P) infrastructure, a next generation version of the earlier architecture, based on loosely coupled administrative relationships and light-weight protocols for unification. The new architecture allows the federation to play a stewardship role and also enables the world to better organize and integrate all climate knowledge via a cooperative federation. With that said, the ESGF P2P software has distinguished itself from other collaborative knowledge systems in the climate community, as evidenced by its widespread adoption, federation capabilities, and broad developer base. It is the leading source for current climate data holdings, including the most important and largest data sets in the global-climate community.

Through ESGF, users access, analyze, and visualize data using a globally federated collection of networks, computers, and software. In addition, an enterprise system was adopted to support international intercomparison activities of climate models and observations as well as high-profile projects involving such organizations as the Department of Energy (DOE), the National Oceanic and Atmospheric Administration (NOAA), the National Aeronautics and Space Administration (NASA), and the National Science Foundation (NSF).

## 2. Prior work and paper organization

This paper is an expanded version of the peer-reviewed article presented at the 2012 IEEE International Conference on eScience [2]. Building on that work, we include here a more detailed description of some of the major software components of the system, in particular the search services, the security infrastructure, the web portal user interface, the model metadata display, data analysis and visualization, and others. We also expand on the survey of other similar infrastructures for eScience, and report on our collaboration with some of these efforts. Finally, we present recent or upcoming developments in the overall architecture and new functionality, such as the next generation of publishing services, the new computing platform, and the plans to apply this infrastructure to other scientific domains.

The rest of the paper is organized as follows: Section 3 describes the high level system architecture; Section 4 presents the major components of the software stack in more detail; Section 5 contains a real life scenario describing how the system may be used by a scientist, and what advantages it represents with respect to previous systems and tools; Section 6 lists the many projects and environments where the ESGF infrastructure is used operationally, or where it will be deployed in the near future; Sections 7 and 8 report on feedback from the user community, as well as lessons learned throughout the project development to date; Sections 9 and 10 compare ESGF to other existing systems, and report on collaborations with those systems or other projects; lastly, Sections 11 and 12 describe short and medium term development tasks, as well as longer-term plans for expanding the system functionality for climate sciences, and generalizing ESGF to other scientific domains.

## 3. Architecture

ESGF is based on a *distributed* and *federated* software architecture (see Fig. 1). The system is composed of multiple sites (called "Nodes") that are geographically distributed around the world, but can interoperate because they have adopted a common set of services, protocols and APIs. Data and metadata are managed and stored independently at each Node; yet, clients can access them as if they were held on a single global archive. Nodes can join or leave the federation dynamically, and the services and holdings that are discoverable and accessible by clients will automatically change to reflect the latest state of the system.

Internally, each ESGF Node is composed of a set of services and applications that collectively enable data and metadata access and user management (see Fig. 2). The software stack includes components that were developed directly by ESGF (e.g., the Publishing program, many of the data and metadata web services, and the web portal user interface), as well as open-source, third-party applications that are commonly used in industry (Postgres, Tomcat, Solr), eScience (MyProxy, GridFTP) and the geospatial community (THREDDS, Live Access Server (LAS)). As such, the full ESGF infrastructure can be considered as a "system of systems", i.e. the result of a careful integration of existing technologies, which have been used reliably by the community to execute specific functions, with

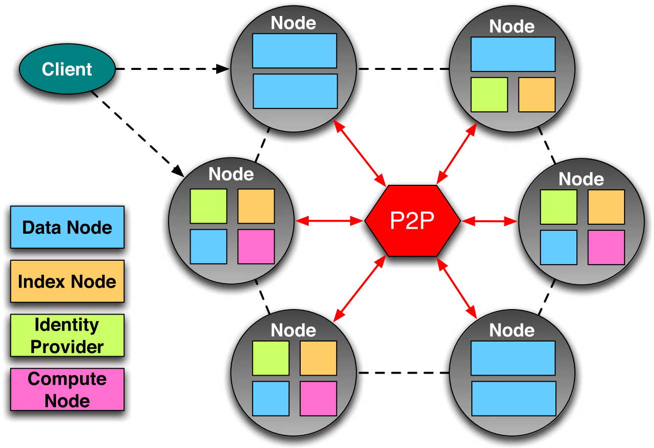



**Fig. 1.** ESGF architecture. The picture provides a schematic representation of the Earth System Grid Federation (ESGF) as a system of Nodes hosting services of four possible "flavors". Nodes continually exchange messages with each other through the peer-to-peer (P2P) protocol (here pictured as the central core of the system). A client can query any Node to access data and metadata throughout the federation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

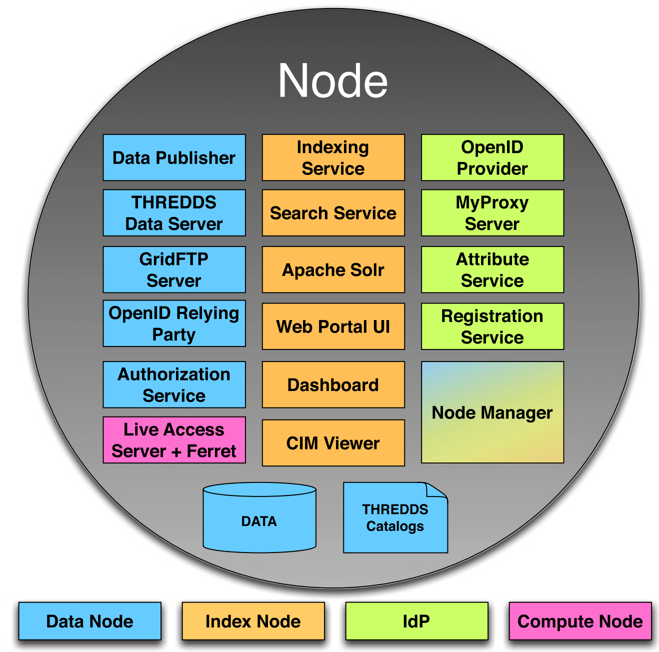



**Fig. 2.** ESGF Node software stack. The software components that constitute the ESGF Node software stack are logically grouped by "flavor" or area of functionality. Each flavor can be installed separately, and flavors can be deployed in multiple combinations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

new software components and middleware to realize a system that covers a wide range of functionality.

The ESGF services are logically grouped in four areas of functionality, which determine the Node "flavors" (color coded in Figs. 1 and 2):

- The Data Node (blue color, Fig. 2) includes services for secure data publication and access. Its main components are the data Publisher application that generates the metadata catalogs, the THREDDS and GridFTP servers (with security filters at the front end) to serve the data, and the OpenID Relying Party and Authorization Service to ensure proper authentication and authorization.
- The Index Node (orange color, Fig. 2) contains services for indexing and searching metadata, currently implemented using Apache Solr as the back-end server. The Web Portal application provides a convenient browser-based interface for system users and administrators, while the Dashboard contains utilities that monitor and report on the state of the federation. The Common Information Model (CIM) Viewer is a plugin component that retrieves and displays model metadata from CIM repository sources.
- The Identity Provider (light green color, Fig. 2) allows user authentication and secure delivery of user attributes. It includes an OpenID Provider for browser-based authentication, a MyProxy server for requesting limited-lifetime user certificates used for non-browser based access, and User Registration and Attribute Services for distributed access control.
- The Compute Node (purple color, Fig. 2) contains higher-level services for data analysis and visualization. Currently its only components are the Live Access Server, the Ferret-THREDDS Data Server and the Ferret engine. More analysis engines (e.g., Ultra-scale Visualization Climate Data Analysis Tools (UV-CDAT)) are to be added in the near future.

Additionally, each of these flavors always includes the Node Manager, a web application that allows a Node to continually exchange service and state information with all its peers throughout the federation. Each Node can be configured during installation to have one, more, or all of the flavors, depending on the site specific needs. Furthermore, a site can combine multiple Nodes configured for different flavors, to achieve higher performance and availability. For example, a common practice is to deploy the Index Node and Identity Provider services on one server, while keeping the Data and Compute Node services on one or more separate servers (e.g., for performance considerations).

ESGF Nodes interoperate through a set of common federation services and mutual trust relationships. A user registered at a given Identity Provider can use this account to authenticate (via OpenID or a short-lived user certificate) with any Node across the entire federation because all Nodes trust the original Identity Provider. A metadata search initiated at any Node will be distributed to all the other Nodes, and the search results will be merged before they are presented to the user. When data is accessed from a specific Node, its local Authorization Service will query the appropriate Attribute Services across the federation, to determine whether the user has been granted the necessary permissions to satisfy the local access control policies. The list of trusted Identity Providers, their Certificate Authorities, the target search Indexes, and the location of the Attribute Services are continually updated at each Node through the peer-to-peer messaging executed by the Node Manager. The net result of this setup is that a client (web browser or desktop application) can query any Node to investigate and access data and metadata holdings distributed throughout the federation.

## 4. The ESGF software stack

ESGF software components are developed collaboratively by many institutions worldwide, which are funded by different agencies and programs. For the U.S., the DOE Office of Biological and Environmental Research is the lead funding agency. The software is made freely available on the ESGF *github* web site (http://esgf.org/gitweb, soon to be replicated to https://github.com/ESGF) under the Open Source BSD license, and it is documented for users, administrators and developers on the ESGF collaboration site (http://www.esgf.org/). The general development philosophy is to build modular components that interact with each other and with outside clients through well-defined interfaces, so that different implementations may be plugged into the system, irrespective of the implementation language (Java, Python, etc.). When possible, ESGF uses existing standards from industry (e.g., OpenID, SSL, SAML) or the geospatial community (e.g., NetCDF, CF, OPeNDAP) to maximize interoperability with other systems. Popular, freely available application servers such as Tomcat, Solr, THREDDS and LAS are integrated into the stack to avoid re-implementing a specific piece of functionality.

In this section, we describe in more detail the major components of the ESGF software stack.

### 4.1. The node manager

The Node Manager is the primary coordinating software component that manages the local ESGF Node and maintains the federation's cohesion. It is an event-driven system composed of 'components' that comprise a SEDA [3] inspired *finite-state machine* (FSM). Events are routed through an arrangement of components and are transformed as they traverse the FSM from start state (ingress) to end state (egress). The main components of the Node Manager implement the discovery and information dissemination protocol that keeps the federation connected and they provide the necessary routing information used by higher-level Node services. The Node Manager is also responsible for collecting current system *health* information – i.e., system metrics – and monitoring information such as cpu, memory, and up-time information as well as quantitative information on a local Node's data holdings. The Node Manager implements *intra*-node message passing through its internal components and enables *inter*-node message passing by way of its peer-to-peer protocol implementation.

### 4.2. The peer-to-peer protocol

A key feature of ESGF is the ability for Nodes to *discover* each other and therefore create a single federated *dataspace* [4] that spans all participating ESGF Nodes. The resultant overlay mesh network created by the ESGF protocol allows users to interact with this singular dataspace to seamlessly query and access data. At its core, the protocol is a *gossip* protocol [5] with two primary purposes: as the *discovery mechanism* allowing Nodes to learn about each other in an unsupervised way, and as the *anti-entropy* [5] protocol that maintains a distributed 'database' of currently federated, participating nodes. The implementation of the gossip protocol in ESGF follows the *push* model [4]. Each Node views the federation as a list of other Nodes it is authorized to access, which is essentially the 'database' maintained via the anti-entropy protocol. This list is encapsulated and sent as payload within a *remote event* that is forwarded to peer Nodes. Nodes receive remote events and inspect the payload information. The payload is then integrated with the receiving Node's internal list, and that new list is pushed randomly to two other peer Nodes. Nodes propagate state information directly after receiving an ingress remote event. Each Node initially creates a list with a single list entry containing its own time-stamped information. As ingress events are encountered, the introduced list is merged with the local list like a vector clock [6], maximally by time.

A gossip protocol was chosen because of its simplicity, scalability, robustness and reliability "*even in the most demanding settings*" [7]. ESGF leverages the gossip protocol's fault tolerance to create a fully connected overlay mesh network of Nodes with no single point of failure. The federation may be partitioned into *groups*, and a Node may participate in any number of groups. Each group must contain at least one Identity Provider Node to service the security needs of the group's users. Each Node is bootstrapped with a prior knowledge of at least one *default peer*. Nodes maintain an eventually consistent data store of all the Nodes in the union of the peer groups they are members of.

Several protocol optimizations are being studied, including tuning fanout [8] and frequency and taking advantage of a push–pull interaction between Nodes. Our ultimate goal is to optimally decrease database convergence time and network overhead.

### 4.3. Data preparation and publishing

The Publisher application is a Python package that is responsible for scanning data stored on a Data Node and making it available through the system. The program, which can be run from the command line or from a graphical user interface (GUI), can extract metadata from both the directory structure and filenames, and from the content of the files themselves. Files are grouped in logical collections named *data sets* according to project-specific rules specified in the relevant configuration file. All information is stored in a local Postgres database and is used to generate THREDDS/XML catalogs. These catalogs contain the descriptive metadata and the URL pointers to access the data through all available protocols (e.g., HTTP, GridFTP). The THREDDS Data Server (TDS) and Live Access Server (LAS) use these catalogs to deliver data to clients for download, analysis, and visualization.

The Publisher contains prebuilt *handlers* for processing data that are stored in NetCDF format and follow the Climate and Forecast (CF) convention, a prescription for encoding richer metadata relevant to climate sciences. Most of the projects that ESGF has focused on to-date mandate that data be stored as NetCDF/CF (e.g., CMIP5, obs4MIPs, ANA4MIPs). Nevertheless, the Publisher package is extensible, and custom handlers can be plugged-in to parse data that are stored in other formats or are organized on disk according to other specific directory structures.

### 4.4. Search services

The ESGF search module, deployed as part of an ESGF Index Node, is responsible for making the data holdings of one or more ESGF Data Nodes discoverable and accessible by clients. Its two main features are the ability to execute distributed searches across the federation, and the support for a well-defined REpresentational State Transfer (REST) [9] API that insulates clients from the specific query syntax of the back-end metadata repository. Although the software allows for pluggable back-ends, it currently uses Apache Solr as the underlying search engine. Solr is a popular web application which is used in many commercial web sites, featuring high-performance text and faceted searching, geospatial and temporal querying, and partition of searchable metadata across multiple local indexes (*cores*) and distributed servers (*shards*).

The search module makes two high-level web services available to clients (see Fig. 3):

- The Indexing Service is invoked by a local or remote client to parse the metadata content available at some repository (located by its URL). This action ingests it into the back-end metadata storage. Specifically, metadata is indexed into a "*master*" Solr index, which is available only locally and opened to both "*read*" and "*write*" operations. From there, metadata records are automatically copied (via the Solr native replication mechanism) at one-minute intervals to a Solr "slave" server, which is publicly available and configured for "*read*"-only operations. To-date ESGF has focused primarily on parsing metadata from THREDDS catalogs, but support for other repositories such as GCMD metadata is forthcoming. Indexing operations into the "*master*" Solr are subject to authorization according to the local policies established at the Index Node: the client must present a valid certificate belonging to a user that has been granted publishing rights for that specific data collection.

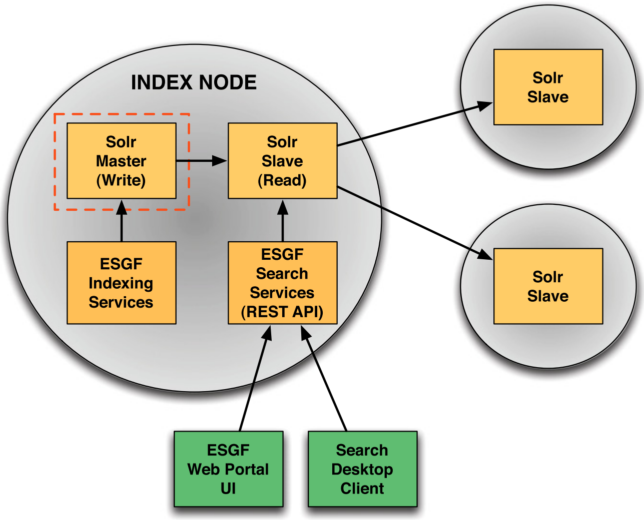



**Fig. 3.** ESGF search services architecture. Each Index Node contains searchable metadata for the local data holdings, and search requests are distributed across all Index Nodes in the federation to return complete up-to-date results to clients.

- The Search Service issues a query to the local "slave" Solr, optionally instructing it to distribute the query to the other entire "slave" Solrs in the federation, and to assemble all the results together before they are returned to the client. Each matching result contains a unique identifier, a descriptive title, and values for all the requested metadata fields (such as the "model", "instrument", "experiment", etc.). Most importantly, results contain hyperlinks to the available data access points (e.g., HTTP, GridFTP, OPeNDAP, and LAS), to allow data download, or invocation of a specific visualization or analysis service. The search service is invoked through a well-defined and documented REST API, which insulates the clients from the more complex query syntax of the underlying Solr server, and that provides higher-level functionality such as automatically distributing the query to all available Index Nodes in the federation, and dropping the un-responsive Nodes from the final response. The search REST API includes the ability to execute free-text queries (e.g. "climate", or "surface temperature"), to search for specific values of one or more search categories or "facets" (e.g. "model = CESM and time frequency = monthly"), to issue geo-spatial and temporal queries, and to combine query clauses in logical "AND" or "OR". Additionally, the API allows clients to search for either the latest or a specific version of the data as well as for its replicas, plus it can target the local index only or any number of remote indexes. Response documents can be returned in either XML or JSON format.

Internally, the search module partitions its metadata space into different object types, which are stored and queried separately—a strategy employed to optimize performance. *Data sets* are the unit of discovery in the system, since they contain high-level information about a collection (such as the model that generated the data, the instrument that collected it, or the collection's temporal and geospatial coverage). As such, data sets are typically returned as results matching a query. *Files* and *aggregations* represent the content of data sets, and contain the URL endpoints of all services that clients can use to access the data. Typically, a single data set contains many files and/or aggregations, so searching for data sets is much faster. Within Solr, each object type is stored in a dedicated "*core*" or index, which is separately updated and queried.

At the time of this writing, the operational ESGF system includes approximately a dozen Index Nodes, distributed across four continents. Benchmarking studies have indicated that distributing a query from one Index Node to another which is located in relative proximity (approximately, in the same continent) takes typically less than a second (depending also on the index size, and query complexity); conversely, carrying a query over the ocean typically results in a waiting time of a handful of seconds (3–4 s). In order to optimize performance and improve the user experience, the ESGF infrastructure has recently evolved to allow Node administrators to create local copies of remote indexes, so that a query issued to the search service at that Node is only executed locally, yet it still spans all metadata available in the federation. Index replication is achieved by leveraging yet again the native Solr mechanism (which has proved to be very reliable), at a configurable time interval, which defaults to 10 min. Each Index Node administrator is free to replicate all remote indexes, a portion of them, or none at all.

Because of their power and flexibility, the ESGF search services have been the building block for the development of additional system functionality. One of these capabilities is the programmatic creation of "wget" scripts to download data files in bulk. These scripts can be created by issuing a request to the ESGF search services that follows the *exact* same syntax as the search REST API, simply changing part of the request URL from "/search" to "/wget". This capability allows a user to first find all data sets matching a set of criteria, and then easily download all files belonging to those data sets. Moreover, the wget scripts contain additional functionality for checking data integrity, resuming interrupted downloads, and re-executing searches which is described in more detail later on in this paper.

Another service built upon searching is the ESGF RSS feeds, which notify subscribed RSS clients about the latest published data sets. Each Index Node exposes several RSS feeds available for subscription: a Node specific feed that informs about data sets published only to that Node, a global RSS feed that spans the whole federation, and a customizable RSS feed that can target a user-selected search category (for example, an RSS feed about all the latest CMIP5 data, across the whole federation). In the future, we plan to expand the ESGF RSS functionality to include multiple search categories, and allow free text content, such as the advertisement of a new relevant scientific result, or the explanation of why a new version of a data set was made available.

### 4.5. Security infrastructure

The ESGF security infrastructure [10] is used to restrict resource access to users who have been granted the necessary permissions, and to manage trust relationships between participating parties across the federation. Permissions can be set at various levels of granularity, such as authorizing a specific user to publish metadata to an Index Node, to download specific data sets from a Data Node, or to manage other user permissions through a Node administrative web interface. For example, access to a data collection might be restricted to only those users who have subscribed to a license agreement (such as the CMIP5 Research or Commercial licenses). Other collections might be limited to a particular research group and not intended for public consumption. ESGF security allows Node administrators to maintain complete control over the local resources, while enabling federation-wide services such as single sign-on and distributed access control. This security framework is accomplished through a distributed network of trusted Identity Providers, Certificate Authorities and Attribute Services.

A typical workflow might involve a user attempting to download some restricted data either through a web browser, or by using a download client such as a wget script (see Fig. 4). To enable access control, the data server (for example, a THREDDS data server) is front-ended with software components that intercept the data request, and enforce proper authentication and authorization by interacting with a system of distributed services for registration,

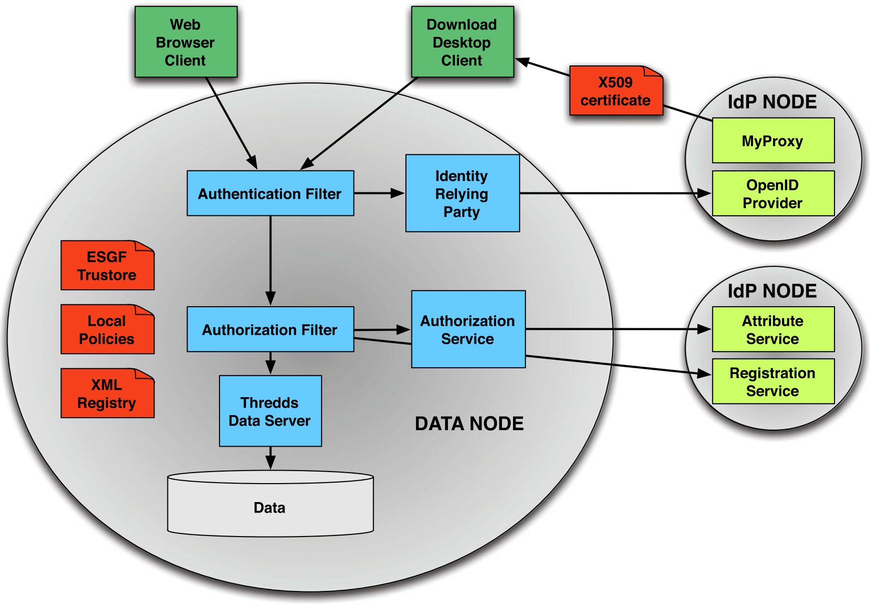



**Fig. 4.** ESGF security example workflow. The user may employ a browser or a desktop client to download data. Server-side components intercept the data request and enforce authentication and authorization by interacting with distributed security services, which may be deployed at other Nodes throughout the federation.

login, retrieval of user permissions, and enforcement of authorization policies.

ESGF enables users and applications to authenticate through two different mechanisms. The *OpenID* protocol is used for interactive user single sign-on through a web browser. When a user enters their OpenID at any Node in the federation, they are redirected to their Identity Provider (the Node where they registered), where they can sign-in using the mechanism enforced by that Provider (for example, username and password). Once authenticated, the user is redirected back to the original Node. *X.509 certificates* are used to enable single sign-on for non-browser based access for example by programs such as *wget* scripts or GridFTP clients. Before executing the program, the user requests a short-lifetime certificate from a MyProxy server that is linked to the user's Identity Provider, and thus to the user's OpenID identity. The certificate contains the user's identity and may also contain other user attributes which can be used to determine access permissions by consuming services. It will be honored by all Nodes in the federation that trust the Identity Provider.

ESGF authorization is based on *security policies*, defined by each Node administrator, that match groups and roles that are allowed to execute specific operations. These groups and roles may be local to the Node or they are global to the federation as in the case of "CMIP5 Research". Users from any site can request membership in one or more access groups (e.g., the "CMIP5 Research" or "NASA OBS" group). In addition, they are assigned one or more roles within each group (e.g., "user", "admin", or "super"). Groups may be managed by any Identity Provider in the federation, and the user attributes may be queried by trusted clients through the Node's Attribute Service. When authorization is required, the local Authorization Service enforces the Node security policies by querying the particular Attribute Service in the federation that manages the configured access control group. For example, any Data Node participating in CMIP5 allows users to access the data if they have enrolled in the group "CMIP5 Research" that is administered by the Lawrence Livermore National Laboratory (LLNL) Identity Provider.

Because ESGF security is based on standards (OpenID, SSL, SAML), the security components can interoperate with trusted components from other software stacks that are based on the same standards. For example, a Java-based ESGF Authorization Service can request and consume user attributes from a remote Attribute Service that is part of a Python software stack, because the entire communication is encoded as SAML/XML documents that have been digitally signed. Similarly, users can authenticate throughout the federation via their OpenID, no matter what the underlying OpenID client and server implementations.

Another key aspect of ESGF security is that it is meant to be non-intrusive, i.e. it can be applied to existing clients and servers without changing their internal code. On the server side, this is accomplished by front-ending existing applications with security filters that intercept the client request (as described in the example above). On the client side, the only requirement is the ability for the client to transmit an X.509 certificate to the server as part of the SSL-handshake in a HTTPS request: this can be accomplished by using standard modern HTTP libraries (available in all the main programming languages), and optionally re-linking the client with data access libraries that are able to automatically locate the certificate on the user's system (such as the latest NetCDF client libraries).

One additional benefit entailed by the pervasive security infrastructure is the ability to log detailed metrics about user requests (i.e., linking each action to a specific user identity). Currently, metrics are collected on a Data Node whenever a user initiates a data download, and then aggregated across Nodes to provide federation-wide statistics. This information can be used to push notifications about data updates and retractions. In the future, logging could be expanded to track other usage of the system such as data searches and administrative actions. Logging has also been used to hold users accountable for intentional or unintentional misuse of the system, such as for example massive data requests resulting in effective denial of service events.

### 4.6. Web portal user interface

A key goal of the current ESGF development effort is to provide an intuitive, easy-to-use web portal. To this end, ESGF has adopted a modern design inspired by emerging Web 2.0 technologies. The portal makes heavy use of the state-of-the-art JQuery

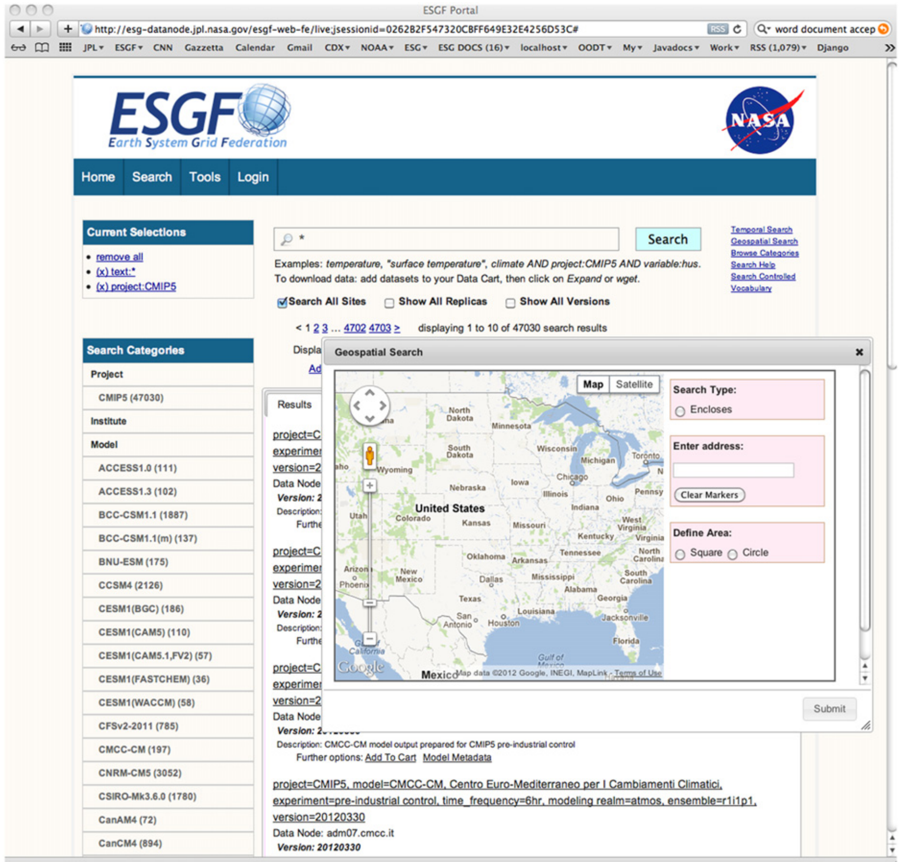



**Fig. 5.** The ESGF web portal. The ESGF web portal exposes a friendly user interface for searching, including the capability to specify free text, facet options, and geospatial and temporal constraints.

JavaScript libraries and Model-View-Controller (MVC) design patterns. The user interface (UI) also conforms to the overall ESGF philosophy of decoupled software components that accommodate a more dynamic user experience. Services of other components (e.g. search and node manager components) may be utilized via Ajax-style HTTP calls. The web UI materializes a variety of ESGF-supported services through the browser: administrative and user account management interfaces, collection and file-level search and discovery, the dashboard service, the LAS visualization engine, and the CIM viewer. The first two features are briefly described in this section, the others later on in this paper.

The *user account and administrative component* delivers a clear, concise means for users and administrators to manage personal information and security aspects of the portal. Users may view and edit personal attributes and apply for group memberships that grant access to restricted data. Administrators are provided with tools that interface to the security infrastructure services allowing them to authorize group access to specific data holdings.

Search and discovery mechanisms are presented to the user via the *search component* of the web portal (see Fig. 5). The search component is presented as a single composite page, which eliminates the clumsy "back-and-forth" navigation offered by static pages. Initially, a user may search the data holdings at the *collection* level. Here, the user has many search options. The sidebar widget on the left side of the page presents a configurable list of *facets*, which allow users to constrain their search results through categorization. Search constraints can also be added via the free text search bar. Furthermore, users may take advantage of the temporal and geospatial (based on Google Maps) search overlay widgets by searching for data measured or simulated at specific times or in specific geographical locations, respectively. The *state* of the search (i.e. the current constraints applied by the user) is shown prominently in the upper left corner of the page and may be changed at any time. Important information about the results is viewed through the portal search component.

Each data set contains a comprehensive metadata summary outlining descriptive information about the data set. The CIM viewer, which provides richer and more detailed metadata about climate models, may also be invoked. Additional information associated with a data set, such as a technical report, may also be accessed. Finally, the LAS browser visualization tool may be invoked to visualize details of the data sets. For file access, the search component uses a "shopping cart" approach commonly deployed by e-commerce sites. The user needs only to select a collection-level data set (via the "Add to Cart" link) and navigate to the "Data Cart" tab. The collections previously chosen are displayed in this tab and may be quickly expanded to their *file*-level data holdings. These files may then be downloaded using different methods, which are discussed in the next section.

### 4.7. Data access

The ESGF infrastructure allows users to download data through multiple protocols and services. The available access mechanisms must be installed by the Data Node administrator on the host where the data are stored, and then specified as valuable access points by the data manager at the time when the data are published. When search results are returned to the user, the data access endpoints are displayed as hyperlinks associated with data sets and files.

By default, every Data Node is configured to serve data through the following access types:

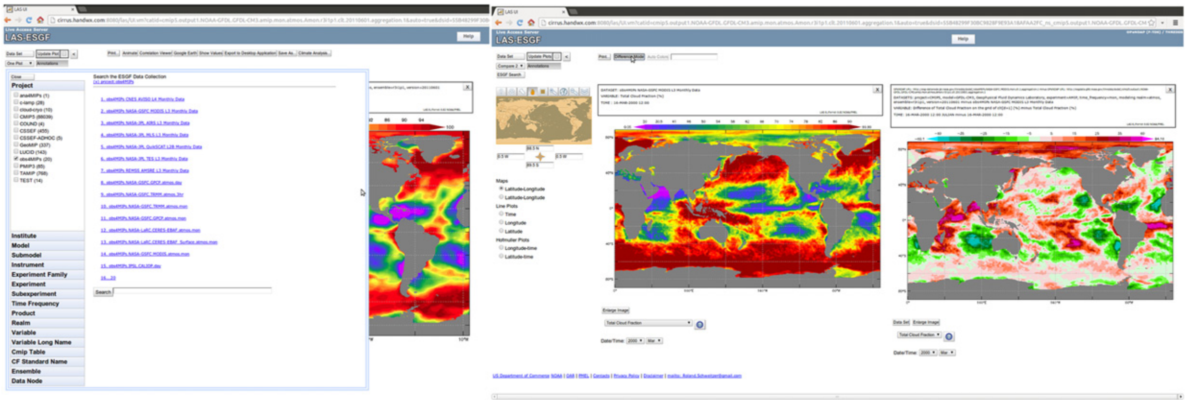


**Fig. 6.** Examples of Live Access Server user interface. The LAS allows users to quickly add data sets to their LAS session (left panel), then compare them on a common grid (right panel). In this case, the plot shows the difference between observation data from NASA JPL and model data from GFDL of the total cloud fraction from March 2000.

- Direct browser access is provided by the TDS and allows users to download full files via simple HTTP hyperlinks.
- OPeNDAP access is also provided by the TDS. The Open Data Access Protocol provides an abstraction level on top of the files allowing users to tailor data selection by variable, temporal or geospatial coverage. ESGF provides aggregations for variables that are normally split in files of no more than 2 GB to simplify access.
- Wget Scripts, contrary to other access mechanisms, are not tied to the Data Node hosting the data, but to an Index Node implementing the search API, which was extended to return a BASH script that simplifies data retrieval. The user can thus access any file in the federation without knowing where the file is stored or even what it is named. The script encapsulates calls to the *wget* utility – hence the name – and adds functionality such as: certificate verification, retrieval and renewal; download continuation, verification via MD5 checksums and verification results caching; download structure generation via the facets results; file overflow check (if the query resulted in a set of files larger than the maximum number allowed per script); and script auto-update and remote file changes check.
- GridFTP access uses a special protocol based on FTP to allow for a high-performance, securely authenticated, and reliable data transfer. Within ESGF, the most common use of GridFTP is for data replication, where bulk data is replicated across Nodes. This feature requires higher access permissions for the user, including write access to the ESGF Data Node, and typically is restricted to administrators.
- Globus Online [11] is a hosted service that provides reliable and managed file-transfer capabilities over GridFTP. The web portal component includes a preliminary integration with Globus Online. A user can choose to have Globus Online either transfer the items in their data cart to another server or download them to their local machines. Users must install Globus Connect on their machines to upload and download data locally.

In the future, ESGF plans to enable other data access mechanisms, and server-side data transformations, for example mapping services based on OGC (Open GIS Consortium) standards to better support regional and local climate studies.

### 4.8. Data analysis and visualization

To-date, ESGF has focused mainly on providing a global infrastructure for indexing and accessing data. As the next step in the project evolution, ESGF is committed to provide powerful analysis and visualization capabilities over these data, on both the server and client side. On the server side, ESGF is already integrated with the Live Access Server engine (described next), and we plan to expand on this capability by connecting to other services, and enabling a wide suite of pre-configured or customizable algorithms to process the data. Server-side computations allow data to be processed where they are stored, thus minimizing network traffic and increasing performance. On the client side, we plan to develop sophisticated tools and libraries that are able to interact with the server-side protocols and APIs exposed by ESGF (such as the search REST API, and the OPeNDAP protocol), so that they can be used to locate the data, download the needed subsets, and execute custom scientific algorithms on the user's desktop. One such tool (the Ultra-scale Visualization Climate Data Analysis Tools or UV-CDAT, described below) already exists; others (such as RCMET, the Regional Climate Model Evaluation System Toolkit, described further in the paper) are in the process of being connected to ESGF.

#### 4.8.1. Live Access Server

The *Live Access Server* (LAS) is a highly configurable, freely available open source web application that allows users to visualize and compare data on-the-fly, perform basic analysis, and download results in various file formats. New capabilities were implemented for reading THREDDS catalogs, managing the data selection UI for large collections, and generating distributed products to ensure a seamless and efficient integration of LAS into the ESGF software stack. In general, LAS can create the configuration it needs directly from THREDDS catalogs and the underlying NetCDF data sources. However, within ESGF, LAS was enhanced to build its configuration from the enhanced metadata in the THREDDS catalogs that are generated by the Publisher application. The web portal user interface allows the user to quickly identify and select the data sets of interest. Once a data set is selected and the user has transitioned to LAS, the user can begin working with the selected data set within the LAS UI. Much of the power of LAS in the ESGF context comes from its ability to do direct comparisons between data from different data sets. In order to make this as seamless and quick as possible for the user, the LAS contains a built-in user interface to the ESGF search API. By selecting this user interface component the user can quickly find and add data sets to their LAS session (see Fig. 6(a)).

Once the data sets of interest have been added to the LAS session the user can choose to display 2 or 4 plots at the same time. The sub-set of the data in the plane of the plot is controlled in the left-hand navigation area and is the same for all data sets. The axes of the data set which are orthogonal to the plot plane are controlled by widgets in the individual plot panels. In this way, users can compare different slices of the same data set (for example looking at two different time steps of the same data).

Users can also choose to place different data sets into each plot window. The controls in the plot plane remain the same making

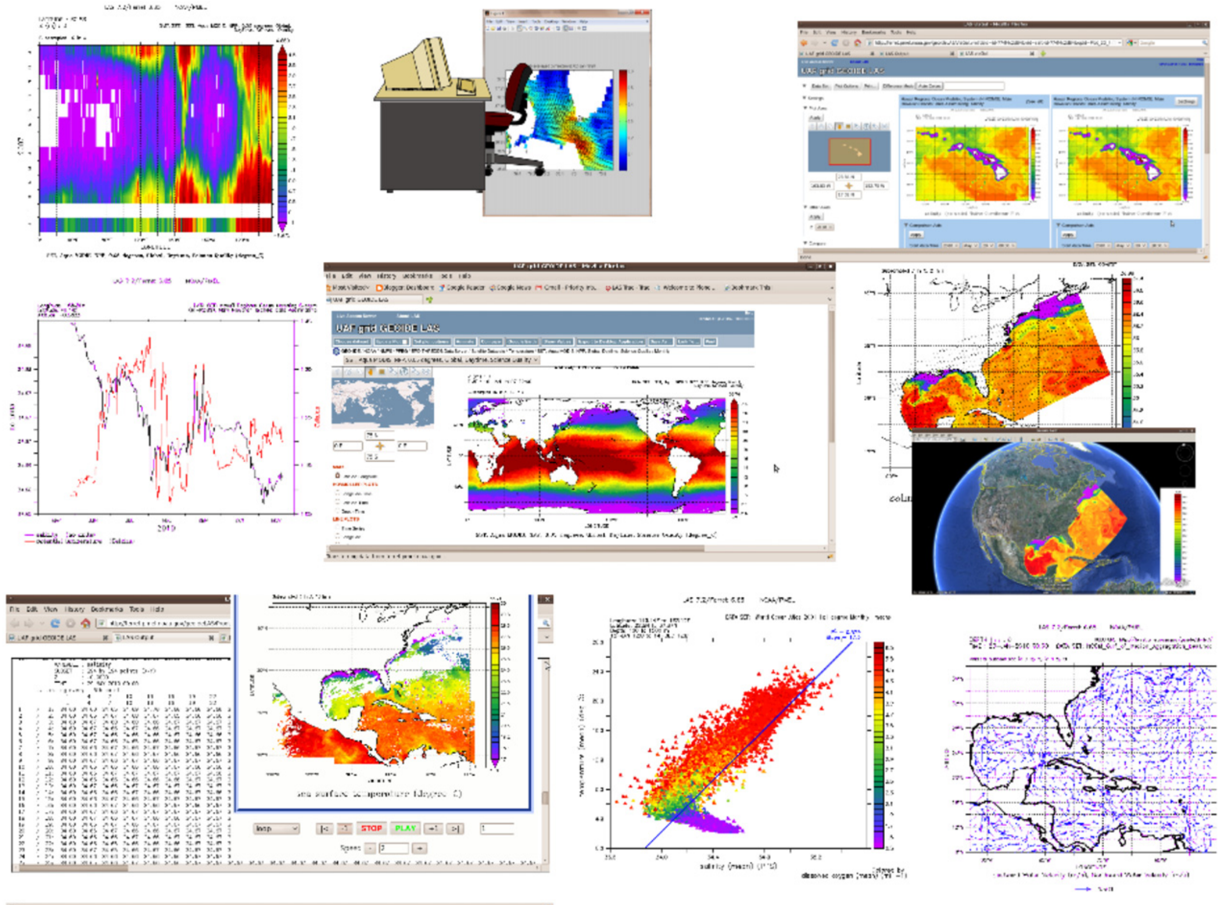


**Fig. 7.** Live Access Server sample plots. The figure shows a collage of some of the science plots that can be produced by LAS over data served through ESGF.

it easy to make direct comparisons between data sets. If the data sets do not overlap in the plane of the plot, a blank plot is displayed for the second data set. In addition to being able to automatically normalize the color scale between all of the plots, users can make direct comparisons by calculating and plotting the difference between the values in the upper left panel with the values in each of the other plots (see Fig. 6(b)). If the data to be differenced are stored on different grids, LAS sends a special OPeNDAP URL to the server hosting the data, which causes the Ferret-THREDDS Data Server to re-grid the data onto the grid of the data set in the upper left of the interface. By executing the re-gridding calculation on the compute node where data are stored, the selected data sets can be directly compared.

Plots and comparisons can be made along any plane or line of the data set allowing users to explore slices of the data along the dimensions relevant to their work (like plotting a time series at a point or the difference of two time series from different data sets at a point). LAS is a rich and fluid environment for exploring earth science data and has capabilities too numerous to detail further, but a flavor of what is possible is visible in the sampling of plots in Fig. 7.

#### *4.8.2. Ultra-scale visualization climate data analysis tools*

In collaboration with multiple U.S. funding agencies (i.e., DOE, NASA, NOAA), universities, and private companies, the Ultra-scale Visualization Climate Data Analysis Tools (UV-CDAT) [12], primarily funded by DOE's Office of Biological and Environmental Research, is developing and providing climate data analysis and visualization capabilities for the ESGF. In the past three years, this large analysis and visualization effort has brought together a wide variety of industry standard scientific computing libraries and applications for the benefit of the climate community. To support interactive analysis and visualization, all components connect through a provenance API to automatically capture meaningful history and workflow. Components can be loosely coupled into the framework for fast integration or tightly coupled for greater system functionality and communication with other components (see Fig. 8(a)). This framework addresses many challenges and opportunities in analysis and visualization of distributed large-scale data sets in the ESGF enterprise system.

United by the standard common protocols and APIs, UV-CDAT integrates over 50 different software libraries and components, of which, the primary software stack comprises of: the Climate Data Analysis Tools, VisTrails DV3D, ParaView, VisIt, Earth System Modeling Framework (ESMF), and R. Its many capabilities include: parallel streaming statistics, analysis and visualization pipelines, optimized parallel input/output (I/O), remote interactive execution, workflow capabilities, and automatic data provenance processing and capturing. It also calls for a novel graphical user interface (GUI) for scientists that include workflow data analysis and visualization construction tools, and the ability to easily add custom functionalities. These are enhanced with the VisTrails provenance tool, the R statistical analysis tool, and state-of-the art visualization (e.g., VTK, DV3D, ParaView, and VisIt), which are brought together within a Python/Qt based GUI (see Fig. 8(b)). While there are overlaps in the functionality of each component, no one component presently addresses all the challenges. By offering multiple components fulfilling similar needs, UV-CDAT caters to the taste of a wide range of users, with different and unique preferences.

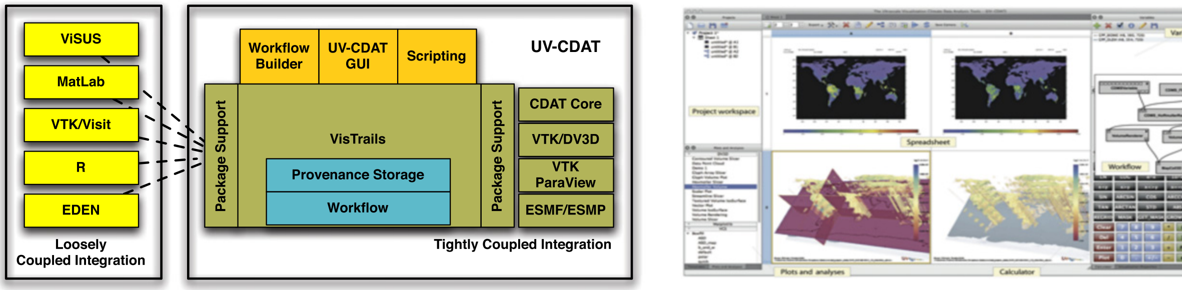



**Fig. 8.** The UV-CDAT client for the ESGF system. Image (a) shows the UV-CDAT's framework component architecture, either tightly or loosely coupled integrated. From this design, other packages can be joined seamlessly or existing packages can be removed without harm to the overall system. Image (b) shows UV-CDAT's main window: project view (top left), plot packages (bottom left), visualization spreadsheet (middle), variable view (top right), workflow (middle right), and calculator (bottom right). For more information on UV-CDAT, visit http://www.uv-cdat.org/.

As climate data sets continue to expand in size and scope, the necessity for performing data analysis where the data is co-located (i.e., server-side analysis) is becoming increasingly apparent. UV-CDAT has therefore undergone modifications to allow ESGF's infrastructure to access its capabilities on the back-end as well as from the front-end. These modifications allow users to not only access the ESGF petabyte archives from their laptops, but also to perform analysis and data reduction before moving the data to their site for further UV-CDAT manipulations. More importantly, the necessary remote operations will be routinely performed, thus freeing UV-CDAT users to concentrate on scientific diagnosis rather than on the mundane chores of data movement and manipulation.

Another use of UV-CDAT is model development and testing. Using 3D slicing through time and space it is possible to isolate systematic errors in both forecast and climate simulations because the user can visualize time and space at the same time. This unique view enables the researcher to see model errors grow and allows first glimpses of model error attribution. This technique along with others are used in the DOE sponsored Climate Science for a Sustainable Energy Future (CSSEF) project where the primary goal is to develop new climate-modeling methodologies to reduce uncertainty through the provision of a large-scale diagnostic test bed—in which UV-CDAT and ESGF are the key software components of the CSSEF test bed architecture.

### 4.9. Dashboard

The Dashboard is the distributed monitoring system of the ESGF. Starting from a prototype developed in the context of the Climate-G research project in 2009 [13,14], the Dashboard framework has been conceived and implemented to address the requirements coming from the Earth System Grid Federation.

Its primary goal is to collect, store and manage monitoring information about the status of the federation in terms of:

- *Network topology* (peer-groups composition)
- *Node type* (host/services mapping)
- *Registered users* (including their Identity Providers)
- *Downloaded data* (both at the Node and federation level)
- *System metrics* (e.g., round-trip time, service availability, CPU, memory, disk, processes, etc.)

From an architectural viewpoint, the ESGF Dashboard is composed of the following three parts: the information provider, the dashboard catalog, and the user interface.

The *information provider* represents the system's back-end and is responsible for retrieving and storing the peer-to-peer metrics. By design, it interacts in a bi-directional manner with the node manager to:

(a) retrieve (and store in the dashboard catalog) updated snapshots of the federation's status, through the XML documents (registration.xml) sent across the P2P network by the node manager. The internal module responsible for performing this task is called RegParser, a XML parser based on the registration.xml schema. The parser exploits lock management and transactions to properly manage the access to the two shared resources registration.xml and node database. This part is already fully operational in ESGF.
(b) provide updated metrics to the node manager (e.g. service availability) to enable a fine tuning of the P2P protocol. This part is currently in a pre-production stage.

The information provider includes also a module called Metrics Collector, responsible for collecting both global and local metrics. This means that each node in the system is able to keep both (i) its own global snapshot about the status of the entire federation (for examples in terms of services availability and network connection status) and (ii) a local snapshot related to its own status (for instance in terms of CPU, memory, disk and processes metrics). From an implementation point of view, the metrics collector is a multithreaded module written in C programming language.

This module can be easily extended to collect new metrics, by implementing additional sensors associated to new classes of information. Apart from system-based metrics, the Metrics Collector extracts and stores information related to the node types and the registered users (at production-level) and the downloaded data (at pre-production level). Additional information on these metrics are out of the scope of this paper and will be presented in a future work.

The *dashboard catalog* is a temporal database (running on top of a Postgresql RDBMS) used by the information provider to persistently store the five classes of information previously mentioned. It manages historical information with a *per-site* and *per-metric* configurable time-window.

Finally, the *user interface* (see Fig. 9) is a web application that exploits the MVC design pattern. It relies on a strong adoption and implementation of Web 2.0 concepts such as mash-up, Google maps and permalinks. It provides several views at four different (hierarchical) granularity levels: federation, peer-group, host and service. These views provide charts and summaries about the service availability, the round trip time between pairs of nodes, the host status, and other information. The views, which can be customized (at the presentation level) by the end user, are delivered in different ways (charts, tables) and formats (XML, CSV, JSON), and can be easily exported (both with and without authentication) in other web-applications, by exploiting the permalink feature implemented into the Dashboard module. From an implementation point of view the Dashboard GUI is written in Java for the back-end and JSP/JS for the front-end. Future work will include moving towards a pure JS object-oriented front-end module and a new set of

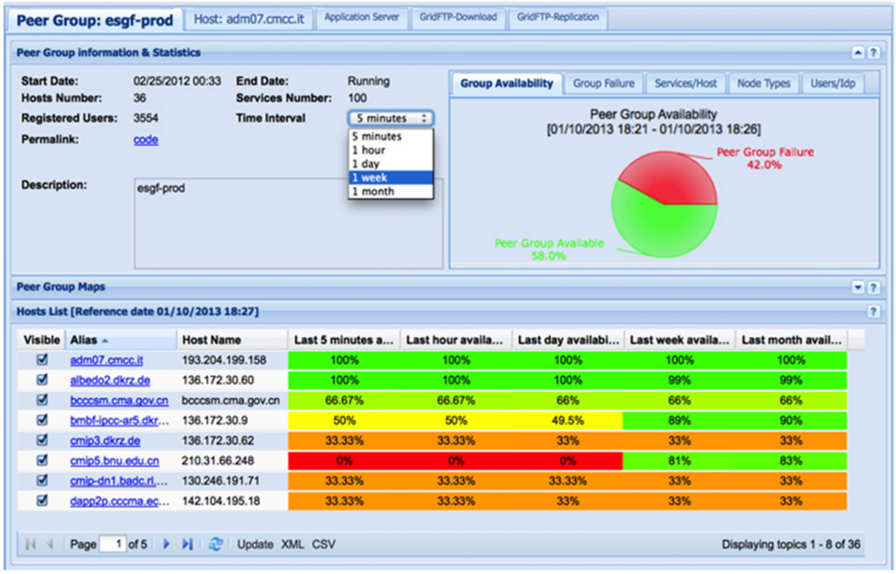



**Fig. 9.** Dashboard web page. The picture shows one of the pages generated by the dashboard module, indicating composition and status of all nodes in the "esgf-prod" peer group.

RESTful web APIs to make all of the metrics available to external applications (other node components, schedulers, new UI, etc.). The last point in particular will address programmability and will allow the Dashboard to primarily act as a "metrics provider" web service.

### 4.10. CIM viewer

Earth System Model (ESM) documentation has traditionally been an afterthought and has certainly hitherto never been standardized. The Metafor Common Information Model (CIM) [15,16] ontology is the first attempt as such a standardization. The Earth System Documentation (ES-DOC) project is taking responsibility for supplying an eco-system built upon the CIM standard. Such an eco-system is designed to support the full workflow of ESM documentation, i.e. creation, validation, dissemination, comparison and review. Following service orientated architectural principles, ES-DOC provides:

- A meta-programming framework that forward engineers both CIM classes and associated XML/JSON decoders/encoders;
- A secure web service CIM API in support of ESM documentation publishing, search and comparison;
- Client tools delivering user functionality such as documentation search, view and comparison.

The ESGF web front-end leverages one such client tool, the lightweight ES-DOC javascript CIM plugin. By leveraging the plugin, the ESGF web front-end is augmented with the functionality to display to the user data set related ESM documentation (see screenshot below). Integrating the plugin into the ESGF web front-end was extremely simple and smooth, an important factor when trying to incubate an eco-system. The ESGF web front-end simply passes to the plugin a data set identifier, the plugin then:

- Invokes the backend CIM API (which returns JSON formatted documentation);
- Transforms the JSON encoded CIM documentation into HTML augmented by javascript.
- Displays the documentation within the user's browser.

ES-DOC also provides ESM comparison tools. Currently such tools support ESM component property comparison. At some point in the near future ES-DOC aims to leverage the ESGF search API in order to harvest search facets that can be used in other comparison scenarios (Fig. 10).

## 5. Example use case

A typical use case scenario that exercises the ESGF system is that of a climate scientist who wishes to analyze the predicted change in some physical field (for example, the "Sea Surface Temperature") in the next 20 years under different emission scenarios. The scientist would start his workflow at any of the ESGF P2P Nodes, and use the search interface to select monthly average data sets from the CMIP5 project that contain the field in question. Because of the distributed nature of the system, the search results would contain data sets that are stored across four continents and tens of Data Nodes. To reduce the number of results returned, the scientist could additionally select a specific model (e.g., "GFDL-ESM2M") and/or a specific emission scenario (e.g., "1% CO2 increase"). Each data set result would contain a link to the LAS server, which the scientist could click to visualize the field over the surface of the Earth, or create a line plot versus time, latitude or longitude. Another link would display the detailed metadata describing both the model and the experiment that were used to generate the output. Once the scientist decided that he indeed located data relevant to his research, he could either download the files from the browser interface (if only a few), or click on the "wget" link to generate a script that he would run on his desktop to download all the files at once. At this point, he would be able to use his analysis package of choice (e.g., UV-CDAT, IDL, MatLab etc.) to take a more in-depth look at the data. Later, the scientist might decide to expand his analysis to include more models and/or scenarios. Instead of clicking many times on the browser UI, he could leverage the ESGF back-end web services to write a simple program that loops over all the models and experiments, generating constrained URLs that can be invoked to obtain the wget scripts programmatically. Additionally, the scientist could decide to compare the model predictions to the current available observations. To do so, he would just need to execute a new search that contained the exact same parameters, but requested data from the obs4MIPs project. The observational data sets could be downloaded, visualized and analyzed just like the model output.

Overall, this example use case shows how the ESGF system enables unprecedented capabilities for scientists to discover and access data, which were either completely unavailable before, or would have taken weeks if not months to perform across disparate and disconnected systems.

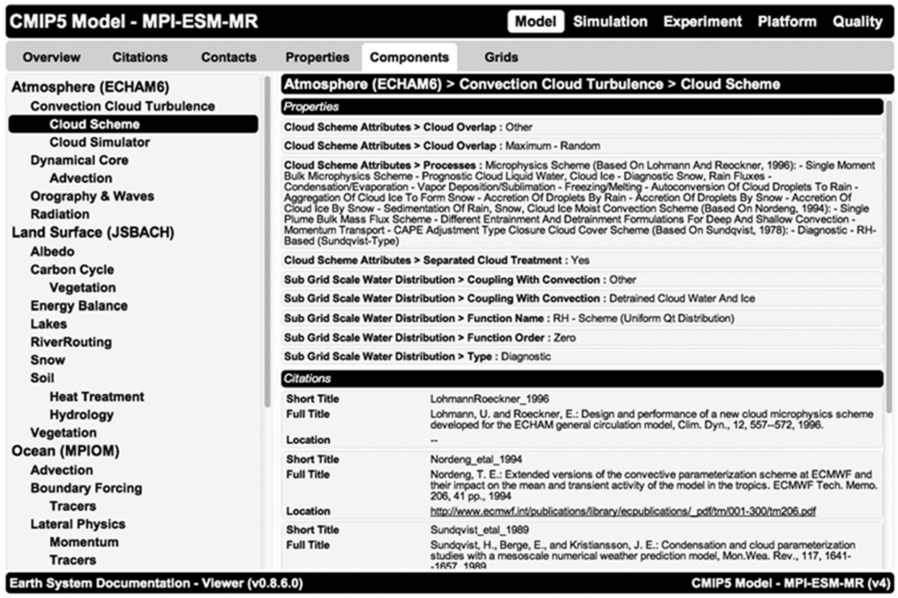



**Fig. 10.** ES-DOC CIM Viewer. The selected panel displays the full ESM component tree plus associated properties and citations.

## 6. Application and operation

From the beginning, the primary focus of ESGF has been to provide the IT infrastructure in support of CMIP5, a global effort coordinated by the World Climate Research Programme (WCRP) that involves tens of modeling groups spanning 21 countries. CMIP5 designed several decadal and century experiments that all models run to assess the future state of the Earth's climate and ecosystem and to understand the complex processes in play during simulated changing climate. These experiments include economic and human development scenarios, Green House Gases emissions and/or concentrations, idealized experiments, and quantifying the predictability of the climate system at the decadal scale. When completed, the CMIP5 data archive will total upwards of 3.5 petabytes, stored across 15 Nodes on four continents (North America, Europe, Asia, and Australia). In order to facilitate access and to secure the long-term preservation of this invaluable data store, ESGF plans to replicate 1.5–2 petabytes of data at five major data centers: LLNL (U.S.), BADC (UK), DKRZ (Germany), ANU (Australia), and the University of Tokyo (Japan).

Alongside the CMIP5 model output, ESGF provides access to the obs4MIPs data sets. These are satellite observations from several NASA missions that have been purposely formatted to allow easy comparison with models: they have been converted from the original HDF/EOS format to NetCDF/CF, averaged as monthly means, and regridded over global grids from the original satellite swaths. In addition, the system provides access to scientific Technical Notes, authored by the data processing engineering teams in collaboration with the mission scientists. These Technical Notes contain invaluable information on how to use the observations for comparing and validating models.

obs4MIPs also includes the CFMIP-OBS data set, which contains observational data and diagnostics about clouds and cloud-related processes. These data can be of great value for climate prediction over a wide-range of space and time scales: for numerical weather predictions, for climate prediction at seasonal to decadal timescales, and for the prediction of the long-term climate response to anthropogenic perturbations at the global and regional scales. This stems from the strong interaction of clouds with the global Earth's radiation balance, the local energy balance, the atmospheric circulation and the hydrological cycle.

Another such effort in support of CMIP5 is ANA4MIPs, which allows users to find and download reanalysis data via ESGF. Through reanalysis climate scientists are continually updating gridded data sets that represent the state of the Earth's atmosphere and incorporating weather observations and numerical output of weather predictions. To enhance climate science resources, five participating reanalysis institutions have agreed to host the following primary reanalysis data sets on ESGF Nodes: (1) MERRA (NASA/GSFC), (2) CFSR (NOAA/NCEP), (3) Interim (ECMWF), (4) 25 (JMA), and (5) 20CR (NOAA/ESRL). These reanalysis data will be translated to the NetCDF/CF format for greater CMIP5 compatibility.

Additionally, CMIP5 has become the benchmark standard that other similar Earth science projects have leveraged in order to make their data accessible to users under a distributed and integrated infrastructure. Some of the most prominent examples are:

- CORDEX: The COordinated Regional climate Downscaling EXperiment is a WCRP-sponsored program aimed at producing an improved projection of regional climate change worldwide for input into impact and adaptation studies. CORDEX will use the multiple-forcings general circulation models (GCMs) from the CMIP5 archive to generate an ensemble of dynamic and statistical downscaling models. The amount of data managed by the system is expected to be comparable to that of CMIP5.
- TAMIP: Transposed-AMIP is a WCRP endorsed activity to run climate models in weather-forecast mode. This activity will primarily focus on analyzing the accuracy of climate models by comparing hindcasts run by those models with observational data.
- GeoMIP: The Geoengineering Model Intercomparison Project is a WCRP-endorsed project aimed at analyzing the effects of manipulating an environmental processes by implementing one or more large-scale geoengineering technologies.
- PMIP: The Paleoclimate Model Intercomparison Project is a WCRP endorsed project designed to evaluate how accurately climate models reproduce climate states that are radically different from those of today. The third phase of this project (PMIP3) shared some experiments with CMIP5 but proposes an additional set of paleo-climate experiments.
- LUCID: The Land-Use and Climate Identification of robust impacts project explores how robust changes in land cover – that

is, those above the noise generated by model variability – affect model output.

All these projects rely, at least partially, on the CMIP5 archive. By providing their results over the same platform they guarantee that users can find, use, and share their data in a standardized manner, thus minimizing the amount of time employed in non-core activities.

Other non-CMIP5 projects that leverage ESGF include:

- CSSEF: Sponsored by DOE's Office of Biological and Environmental Research, the Climate Science for a Sustainable Energy Future project is developing a new climate-modeling methodology to reduce uncertainty through the provision of a test bed.
- CMDS: The Coastal Marine Discovery Service is a NASA-sponsored activity aimed at providing Global Information System visualization capabilities for prominent OPeNDAP-accessible oceanic data sets.

## 7. Usability, performance and reliability

At the time of this writing, the ESGF P2P system has been under development for approximately three years, and deployed operationally for almost two years. It is now use daily by a large community of scientists and researchers around the world.

Users have in general been very appreciative of the new system, praising the responsiveness, completeness and consistency of the search results across Nodes in the federation. In particular, the so called "power-users" are making extensive use of the back-end search services that allow them to query and generate data download scripts programmatically for different sets of constraint parameters. Other facilities such as the LAS, the CIM Metadata Viewer, and the Dashboard are providing users and administrators with functionality that simply did not exist before.

One early problem detected on the system was the excessive memory usage of the TDS when indexing very large collections of thousands of catalogs, which made the data-download services unavailable for long periods of time. This problem seems to have been solved with the latest TDS release. Other aspects of the system that demanded particular attention, but has become progressively more reliable as the system evolved, include the coordinated installation of the several software components, the federated security infrastructure, and the client-side support for data download (i.e., the wget scripts, which must be made to work on a large variety of user operating systems).

Because the system is globally distributed, software integration and updates can be challenging. Time zone coordination across modeling and data centers must be done with the upmost care, as to not disrupt or hinder the scientific community from their research activities. These changes in the system mandate a commitment to backward compatibility for all federated services, as the typical lag time for an update across all Nodes in the federation may span several weeks. It also requires a careful job of testing and documenting each release—a process that admittedly is still under improvement today.

## 8. Lessons learned

The development, management, and support of the ESGF P2P system is a very demanding but rewarding activity for the climate science community. The ESGF software, formally known as the Earth System Grid (ESG) from 1999–2010, has attracted a broad developer base of dedicated individuals from many worldwide time zones. Originally established for the federation of CMIP multi-model archives, the ESGF management communicates closely with the WCRP to establish data requirements and to set strict timelines. Because of our close ties with WCRP and other climate research activities, ESGF has been widely adopted so that it is now being utilized in serving the most comprehensive multi-model and observational climate data sets in the world. Modifying ESGF to meet the requirements of projects that it was not originally intended for is an ongoing challenge, and modifications are underway to accommodate new projects and their data. Consequently, the ESGF infrastructure is under constant requirement to improve and adapt. However, we believe that many of its core architecture principles represent the cornerstone of future systems addressing "big data" challenges across many science domains. From our many years of experience, we describe lessons learned, which may be useful for similar or next generation distributed systems following in our footsteps.

- Technologically, a global infrastructure supporting the data needs of a science domain cannot be conceived by implementing a completely new software stack from the ground up—rather, it must rely on the careful integration of already proven technologies and applications that have been developed by teams over the course of many years, and that are already popular with a large community of users. Also, reliance on standards for data discovery, access and processing is paramount, as it constitutes the basis for interoperability with other systems and clients. In this regard, we think that ESGF has followed the right course from the beginning, by adopting applications engines such as Solr, TDS, and LAS, and conforming to standards such as HTTP, SSL, OpenDAP, and others.
- From the development point of view, we firmly believe in promoting participation and involvement by a large community of stakeholders, managers, and engineers, through an open source, meritocracy based system (not dissimilar to the principles promoted by the Apache Software Foundation, for example). Earlier attempts to restrict development to a selected group of people, and demand contributive license agreements from collaborators, almost caused the project to be derailed.
- As a related item, a common mistake that should be avoided is to develop single points of failure in the engineering workforce. Too often, while developing such large system by a wide collaboration of professionals, a critical part of the system ends up being understood and maintained by a single developer. This situation can lead to serious problems if that developer is temporarily unavailable, leaves the project altogether, or is not open to improve that component based on feedback from others.
- Large software development projects, that span many institutions and funding agencies, should establish a governance model from the very beginning, in order to represent the interests of all stakeholders, prioritize requirements, and guide the overall system development. In our case, ESGF evolved over the course of many years, including changes in the participating institutions, expanded community acceptance, and new areas of applicability. This evolution generated a significant change in scope ("mission creep") from what ESGF was originally intended, which in turn caused disagreements in direction and strife in the entire system development. Only recently ESGF has adopted a governance model that represents the interests of all parties involved, and will be guiding the future evolution of the project.
- These days, a large data infrastructure project should consider scalability as one of its major requirements. All software components that involve data discovery, movement and processing should be architected and tested to scale to 10 or 100 times the current amount of data, because the data collected or generated across all IT domains (including science domains) is growing faster than ever in human history. The project should select scalable technologies in the first place, and continually monitor

and evaluate new solutions as they become available. Designing the system services according to high-level APIs, which become "contracts" between the system servers and their clients, can typically facilitate replacement of existing technologies with more scalable ones.

- One aspect of the project that is still a source of struggle is the difficulty to secure a permanent source of funding to support the system operations, and its evolution into the future. Both U.S. and European agencies tend to fund innovative research and new ideas, but they are less prone to provide ongoing support for projects that have been successful to the point of being the worldwide reference infrastructure in a specific science domain. Funding agencies should consider revising their priorities, or perhaps international organizations such as the WCRP should consider providing support for operational infrastructure.
- Data assets such as storage and compute hardware are physically distributed around the globe and connected by varying speeds of the Internet. These node resources are connected to one another, and to users, to provide search, acquire data, and preform computation. However, most users subscribe to an old paradigm in which they bring subsets or reduced versions of the data they desire to their local machines for analysis. This is restrictive because it uses slow Internet transfers combined with remote computation only as a preprocessing step and does not exploit the capabilities within the analysis workflow, where it could lead to better efficiency, higher speed, and more accurate scientific inferences. For future activities, we are exploring how to leverage new high-performance state-of-the-art compute facilitates and networks to optimize climate use cases for the analysis of distributed and federated data.
- Finally, as in the commercial world, users should come first. A project that aims at being successful on the basis of providing services to a community should focus a large percentage of its resources upon community outreach and user support, such as establishing support email lists, and authoring well conceived tutorials, videos and APIs. Too often documentation and user support is considered as an after-thought, since developers enjoy more working at a new piece of functionality as opposed to making the current system more reliable and easier to use. The project workforce should include both system developers, and system users, i.e. people that are specifically trained in using the system and dedicated to helping others. Also, once the system is operational and has gained a large user basis, any change in functionality must be clearly and promptly communicated to the users. For example, in the recent past the system was improved by changing how bulk download scripts are generated, but this change was not appropriately advertised to the users, with a consequent spike in support requests. In conclusion, ESGF itself should probably dedicate more resources to supporting users, improving documentation, and advertising its capabilities. Unfortunately, this necessity is in stark contradiction with the nature of available funding as described earlier.

## 9. Comparison with existing systems

The ESGF P2P infrastructure is only one of many systems, currently operational throughout the world, that are devoted to enable web-based access to scientific data. Many of these systems have great merits, offering functionality that partially overlaps and sometimes extends that of ESGF. However, ESGF certainly stands out for its federation capabilities, size of data holdings, broad developer base, extended adoption, and most notable data holdings (i.e., CMIP5, obs4MIPs, ANA4MIPs).

A few of the most noticeable examples of other science data systems include:

- The NASA Global Change Master Directory (GCMD) is a well-established clearinghouse for geo-scientific metadata records based on the popular DIF metadata standard.
- The British Atmospheric Data Centre (BADC) has a Python-based infrastructure that shares many technologies with ESGF, and is interoperable based on common APIs and trust relations.
- The NOMADS network of NOAA sites is serving a plethora of heterogeneous Earth data sets via TDS servers and OpenDAP, sometimes enhanced by a Live Access Server.
- The USGS Geo Data Portal provides access to climate data sets and derivative products through standard OGC protocols.
- The Mercury system is a web portal deployed at ORNL that provides extensive metadata capabilities and data access for DOE observations and models.
- The NASA Planetary Data System (PDS) provides access to archived observational data from robotic planetary science missions and other international Space centers, is based on a federation model similar to ESGF, and uses some of the same search technology such as Solr.
- The Group on Earth Observations has developed a ten year implementation plan for a GEOSS (Global Earth Observation System of Systems). This follows a system of systems approach to build a Common Infrastructure comprised of existing infrastructures linked together from around the world to serve Earth observation data. It supports search, and data tools and services accessible through the GEO Web Portal. It currently includes over eighty participating countries as well as the EU.
- The European Space Agency's HMA (Heterogeneous Mission Accessibility) defines a Service Orientated Architecture (SOA) to provide access to data from multiple missions in the context of the European Union's Global Monitoring for Environment and Security (GMES). The architecture makes extensive use of OGC specifications. The HMA-T project established a testbed and HMA will feed into ngEO, the Next Generation User Services for Earth Observation. ngEO also builds on ESA's SSO (Single Sign-on) system, a deployment of Shibboleth and the OpenSearch Geospatial Extension for data discovery.

All these systems were developed largely independently based on different sources of funding, and to fulfill specific institutional and science domain requirements. Despite its current fragmentation, the geo-informatics community is realizing the need for a cohesive approach and increased interoperability to meet the challenges of globalization and ever-increasing data volumes. For long-term survival of ESGF, collaborations with many of these groups and others are well underway. Primarily engagement of all interested parties include an open software engineering process that will promote the definition of common APIs, the adoption of reusable and modular software components, and the expanding suite of capabilities (e.g., Machine learning, visualization, graph analysis, etc.) available to all science communities.

## 10. Collaborations

ESGF is a successful international collaboration that is recognized as the leading infrastructure for the data management and access of large distributed-data volumes in climate-change research. ESGF, which originated in the DOE funded Earth System Grid, has evolved to encompass tens of data centers worldwide, collectively holding tens of petabytes of data, and serving tens of thousands of users. We continue to build upon the widespread success of ESGF in several ways, from its software architecture to its collaborative research and development paradigm, and as a model for supporting software administrators and the user community. ESGF will offer these capabilities, ensuring in collaboration with the other international institutes that they are capable of handling increasing data volumes and adapt well to the changing architectural constraints of systems. ESGF will also support customization of tools to other climate science research, including:

- The Regional Climate Model Evaluation System (RCMES) [17] is an open source, collaborative software effort led by the Jet Propulsion Laboratory, California Institute of Technology (JPL) and the UCLA Joint Institute for Regional System Science and Engineering (JIFRESSE). RCMES is a modular architecture, providing two fundamental components: an elastic, cloud database of remote sensing observations, RCMED, that provides a space/time query and retrieval web service for relevant regional and global observations, and reanalysis data sets; and an analysis toolkit, RCMET, for temporal and spatial regridding; seasonal cycle compositing (all Januaries; all Summer months, etc.), metrics calculation (bias, RMSE, PDFs) and visualization using NCAR NCL and Python APIs to expose it (PyNGL). RCMES is used in support of model evaluation, including providing core capabilities for the CORDEX community, for the U.S. National Climate Assessment (NCA), and for the North American Regional Climate Change Assessment Program (NARCCAP). We are working to develop an integration between RCMES and ESGF, initially focusing on the ESGF search component, and its connection to RCMET for use in model search and download. This collaboration is funded through NASA's Advanced Information Systems Technology (AIST) program (ID: AIST-QRS-12-0002) and through the NASA Computational Modeling and Cyberinfrastructure (CMAC) program (11-CMAC11-0011) and through NASA's contribution to the U.S. National Climate Assessment (NCA) (ID: 11-NCA11-0028).
- OpenClimateGIS (OCGIS) [18] is an open-source project funded by NOAA to develop a Python library for subsetting geophysical data by arbitrary geographic shapes (for example, the continental U.S., or the Colorado river basin), compute derived products such as climate indexes and other second order products, and stream subsetted/computed data into multiple formats. ESGF plans to leverage OCGIS to enhance its support for regional and local climate analysis, and as a framework to execute batch processing of climate model output on the Data Nodes. ESGF and OCGIS are already in close contact for sharing development and ideas.

## 11. Upcoming development

The ESGF infrastructure is in continuous evolution, as the project team strives to meet the demands of additional functionality and higher performance brought upon by an ever increasing community adoption. Some exciting new features are already being worked and will have been released in production by the time this paper is published.

One of the most important upcoming changes is a *next generation publishing framework*, which will enable ESGF to both reach across other systems in the Earth Sciences domain, and to expand its application domain to other scientific disciplines. So far, ESGF publishing capabilities have been limited to parsing NetCDF files organized in THREDDS catalogs, which was enough to support the global CMIP5 effort and related climate projects. In the short future, we plan to develop services to greatly enhance the variety of resources that can be published and discovered through the system, as well as supporting multiple protocols for invoking these services by clients. The new architecture will support publication through both a "*pull*" mechanism, whereby clients request the service to harvest a remote metadata repository, and a "*push*" mechanism, whereby clients send complete metadata records directly to the service for ingestion. The system will support the ability to plug-in new harvesting algorithms (to generate metadata from heterogeneous sources), as well as to specify project or domain specific metadata schemas, which will be used to validate incoming records. For example, ESGF will support the publication of other formats for geophysical data (HDF, GRIB, etc.), other kind of resources (images, documents, etc.), and resources from other domains (biology, astronomy, etc.).

Another upcoming development is the *new computational platform*, which will greatly expand the capability of the Compute Node software stack. It will support parallel and distributed computing tasks by including OpenMPI, Map/Reduce and streaming computing models. The new compute node will allow for large-scale manipulation and analysis of data. These computations will be coordinated within an organizational unit as well as across organizational units directed by data locality. This capability coupled with the current distributed 'dataspace' capability will give ESGF the ability to take advantage of local and remote computing resources to efficiently analyze data across the ESGF federation. Each ESGF participating organization will be able to quickly deploy and setup a compute cluster and provide policies to govern its local resource utilization.

Finally, ESGF is developing capabilities to track provenance of complex data processing workflows for documentation and reproducibility of scientific results. Provenance is defined by the Oxford English Dictionary as the source or origin of an object; its history and pedigree; a record of the ultimate derivation and passage of an item through its various owners. For science, provenance is important because it helps to interpret and reproduce the results of an experiment (such as in the case of CMIP5); help understand the reasoning used in the production of a result; verify that the experiment was performed according to acceptable procedures; and track who is responsible for its results. The Provenance Environment (ProvEn) that is being developed exclusively for ESGF will support scientific provenance that can be browsed as well as queried across multiple workflow and data set histories. The design while general purpose is based on use case drivers taken from the Department of Energy (DOE) Climate Science for Sustainable Energy Future (CSSEF) project whose focus is on the reduction of uncertainty within climate models. The project is developing a sophisticated test bed for model validation and uncertainty quantification within the framework of the ESGF using an extended set of observational data to address global atmospheric, oceanic and land modeling.

## 12. Future directions

While assisting with periodic model assessments, the climate community has observed the CMIP archive rapidly expand from CMIP1 (1 gigabyte (GB)—early 1990s), CMIP2 (500 GB—late 1990s), CMIP3 (35 TB—2007), and CMIP5 (3.5 PB—2013). CMIP6 is expected to hold between 350 PB to 3 exabytes (EB) of data, containing more sophisticated climate simulations and observational data at much higher resolutions. Based on current growth rates, the climate community presents a real need for networks far faster than those available today. To move current and future large-scale data in a decentralized problem-solving environment (such as ESGF), a faster network – on the order of 10 to 100 times faster than what exists today – will be needed to deliver data to scientists when they need it and to permit comparison and combination of large (sometimes 100 s of TB) data sets generated at different locations. These goals cannot be achieved using today's 10 Gigabit per second (Gbps) networks. Therefore we need to ensure that the ESG architecture scales to the next generation network speeds of 100 Gbps and ultimately beyond (i.e., networks beyond 1 Terabit per second). With this goal in mind, ESGF and Energy Sciences network (ESnet) are working closely to build a system that will integrate massive climate data sets, emerging 100 Gbps networks, and state-of-the-art data transport and management technologies to enable realistic at-scale experimentation with climate data management, transport, and analysis in a 100 Gbps, exabyte world.

Recent outreach activities have demonstrated the easy adaptability of the ESGF infrastructure to other relevant domains such as astrophysics, biology, and energy, and we are confident it provides the desired flexibility to support a wide range of other science domains too. While not abandoning the climate community, we are first pursuing the use of ESGF in the biology community, specifically to accelerate drug design and development. ESGF's large-scale data management infrastructure will be augmented with community biological tools for data mining, analysis, and visualization. Moreover, a computational infrastructure will be created to facilitate the exchange of biological data such that validation and comparison with historical data will be easily accessible, and data mining capabilities to help extract knowledge and information will be routinely used to create and enhance new predictive capabilities for drug development. The core ESGF system will allow integration of data from many existing experimental efforts in genome sequence, structural genomics, biochemical kinetics, and pharmacology. By working in other science domains, the ESGF team will gain considerable experience with tools and practices required to support several categories of users—from system administrators who need to install, operate, and maintain the software to a wide spectrum of scientific researchers who want to easily and efficiently cross-reference, access and analyze disparate scientific domain data.

Many challenges remain for climate scientists, and new ones will emerge as more and more data pour in from climate simulations run on ever-increasing high-performance computing (HPC) platforms and from increasingly powerful, higher-resolution satellites and instruments. From these current and future challenges, ESGF envisions a scientific data universe in which data creation, collection, documentation, analysis, preservation, and dissemination are expanded not just to benefit the geoscience community, but also for other scientific domains, such as biology, chemistry, energy, fusion, and materials. By combining with other disciplines, ESGF seeks to strengthen the entire scientific community with advances that will be beneficial for all. In addition to our trademark advances in data preparation, search and discovery, data access, security, and federation – as mentioned in this paper – we will focus our efforts on expanding into the new areas of data mining, provenance and metadata, workflows, HPC, data movement, and data ontology. Furthermore, we intend to fully explore the possibility of providing a configurable and scalable ESGF environment that can be easily deployed on the Cloud, so that resources and services can be instantiated on demand for specific short-lifetime projects, or to meet other requirements such as high availability, extended storage, and elastic allocation of computing processes. As future computing platforms, satellites, and instruments expand and reach extraordinary speeds and levels, we anticipate that in the coming decade ESGF software will migrate from managing petabytes to zettabytes of federated data. To prepare for such an expansion – with continual delivery of a comprehensive end-to-end solution for the overall increase in scientific productivity – ESGF is also partnering with private industry in an effort to make substantial investments in data-driven software and technologies paramount for future computing platforms and ultrascale data archives.

## Acknowledgments

The development and operation of ESGF is supported by the efforts of principal investigators, software engineers, data managers and system administrators from many agencies and institutions worldwide. Primary contributors include ANL, ANU, BADC, CMCC, DKRZ, ESRL, GFDL, GSFC, JPL, IPSL, NCAR, ORNL, LBNL, LLNL (leading institution), PMEL, PNNL and SNL. Major funding provided by the U.S. Department of Energy, the National Atmospheric and Space Administration (NASA), and the European Infrastructure for the European Network for Earth System Modeling (IS-ENES).

A portion of this work was performed by the Jet Propulsion Laboratory, California Institute of Technology under contract to the National Aeronautics and Space Administration.

## References

[1] D.N. Williams, R. Ananthakrishnan, D.E. Bernholdt, S. Bharathi, D. Brown, M. Chen, A.L. Chervenak, L. Cinquini, R. Drach, I.T. Foster, P. Fox, D. Fraser, J. Garcia, S. Hankin, P. Jones, D.E. Middleton, J. Schwidder, R. Schweitzer, R. Schuler, A. Shoshani, F. Siebenlist, A. Sim, W.G. Strand, M. Su, N. Wilhelmi, The earth system grid: enabling access to multi-model climate simulation data, Bulletin of the American Meteorological Society 90 (2) (2009) 195–205.

[2] Luca Cinquini, Daniel Crichton, Chris Mattmann, John Harney, Galen Shipman, Feiyi Wang, Rachana Ananthakrishnan, Neill Miller, Sebastian Denvil, Mark Morgan, Zed Pobre, Gavin M. Bell, Bob Drach, Dean Williams, Philip Kershaw, Stephen Pascoe, Estanislao Gonzalez, Sandro Fiore, Roland Schweitzer, The earth system grid federation: an open infrastructure for access to distributed geo-spatial data, in: Proceedings of the 2012 IEEE Conference on eScience.

[3] M. Welsh, D. Culler, E. Brewer, SEDA: an architecture for well-conditioned, scalable internet services, in: Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles, SOSP '01, ACM, New York, NY, USA, 2001, pp. 230–243.

[4] Michael Franklin, Alon Halevy, David Maier, From databases to dataspaces: a new abstraction for information management, SIGMOD Record 34 (4) (2005) 27–33.

[5] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, D. Terry, Epidemic algorithms for replicated database maintenance, in: Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing, PODC '87, ACM, New York, NY, USA, 1987, pp. 1–12.

[6] L. Lamport, Time clocks and the ordering of events in a distributed system, ACM Communications 21 (7) (1978) 558–565.

[7] W. Vogels, R. van Renesse, K. Birman, The power of epidemics: robust communication for large-scale distributed systems, in: Proceedings of HotNets-I, Princeton, NJ, 2002.

[8] S. Verma, W.T. Ooi, Controlling Gossip Protocol Infection Pattern Using Adaptive Fanout, in: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems, ICDCS 2005, pp. 665–674, June 2005.

[9] Fielding Roy Thomas, Architectural styles and the design of network-based software architectures, Doctoral dissertation, University of California, Irvine, 2000.

[10] Philip Kershaw, Rachana Ananthakrishnan, Luca Cinquini, Dennis Heimbigner, Bryan Lawrence, A Modular Access Control Architecture for the Earth System Grid Federation, in: Proceedings of the 2011 International Conference on Grid Computing and Applications, July 2011.

[11] B. Allen, J. Brenham, L. Childers, I. Foster, G. Kandaswamy, R. Kettimuthu, J. Kordas, M. Link, S. Martin, K. Pickett, S. Tuecke, Software as a service for data scientists, Communications of the ACM 55 (2) (2012) 81–88.

[12] Dean Williams, Charles Doutriaux, John Patchett, Sean Williams, Galen Shipman, Ross Miller, Chad Steed, Harinarayan Krishnan, Claudio Silva, Aashish Chaudhary, Peer-Timo Bremer, David Pugmire, Wes Bethel, Hank Childs, Mr Prabhat, Berk Geveci, Andrew Bauer, Alexander Pletzer, Jorge Poco, Tommy Ellqvist, Emanuele Santos, Gerald Potter, Brian Smith, Thomas Maxwell, David Kindig, David Koop, The Ultra-scale Visualization Climate Data Analysis Tools (UV-CDAT): Data Analysis and Visualization for Geoscience Data, Computer, 27 March 2013, IEEE computer Society Digital Library, IEEE Computer Society, http://doi.ieeecomputersociety.org/10.1109/MC.2013.119.

[13] S. Fiore, A. Negro, G. Aloisio, The climate-G portal: the context, key features and a multi-dimensional analysis, Future Generation Computer Systems 28 (1–8) (2012) http://dx.doi.org/10.1016/j.future.2011.05.015.

[14] S. Fiore, A. Negro, G. Aloisio, The data access layer in the GRelC system architecture, Future Generation Computer System 27 (3) (2011) 334–340. http://dx.doi.org/10.1016/j.future.2010.07.006.

[15] Eric Guilyardi, V. Balaji, Sarah Callaghan, Cecelia DeLuca, Gerry Devine, Sebastien Denvil, Rupert Ford, Charlotte Pascoe, Michael Lautenschlager, Bryan Lawrence, Lois Steenman-Clark, Sophie Valcke, The CMIP5 model and simulation documentation: a new standard for climate modeling metadata, CLIVAR Exchanges 16 (2) (2011) 42–46.

[16] B.N. Lawrence, V. Balaji, P. Bentley, S. Callaghan, C. DeLuca, S. Denvil, G. Devine, M. Elkington, R.W. Ford, E. Guilyardi, M. Lautenschlager, M. Morgan, M.-P. Moine, S. Murphy, C. Pascoe, H. Ramthun, P. Slavin, L. Steenman-Clark, F. Toussaint, A. Treshansky, S. Valcke, Describing earth system simulations with the metafor CIM, Geoscientific Model Development Discussions 5 (2012) 1669–1689. http://dx.doi.org/10.5194/gmdd-5-1669-2012.

[17] K. Whitehall, C. Mattmann, D. Waliser, J. Kim, C. Goodale, A. Hart, P. Ramirez, P. Zimdars, D. Crichton, G. Jenkins, C. Jones, G. Asrar, B. Hewitson, Building model evaluation and decision support capacity for CORDEX, WMO Bulletin 61 (2) (2012) 29–34. http://www.wmo.int/pages/publications/bulletin_en/61_2_cordex_en.html.

[18] Tyler A. Erickson, W. Benjamin, K. Koziol, Arthur Endsley, Richard B. Rood, OpenClimateGIS: a web service providing climate model data in commonly used geospatial formats, poster presented at the American Geophysical Union, San Francisco, CA, December 2011.

**Luca Cinquini** is a senior software engineer at the NASA Jet Propulsion Laboratory, where he is involved in architecting and developing software components for management and analysis of Earth System data. His major activity is working at the Earth System Grid Federation, the global infrastructure used for world-wide distribution of model and observational data for climate change research. Luca collaborates closely with NOAA to promote shared development and common infrastructure between the two agencies. Before being involved in the Earth Sciences, Luca used to work in High Energy Physics, where he got his Ph.D. from the University of Colorado.

**Daniel J. Crichton** is a principal computer scientist, program manager, and principal investigator at the NASA Jet Propulsion Laboratory. His research interests include data-intensive systems for planetary, earth and biomedical sciences. He leads data system development efforts for two large networks in planetary science and cancer research and oversees JPL's earth science data systems program. He has numerous publications and recently served on the National Research Council's Committee on the Analysis of Massive Data. Crichton has an M.S. in computer science from the University of Southern California.

**Chris A. Mattmann** is a senior computer scientist at the NASA Jet Propulsion Laboratory, working on instrument and science data systems on Earth science missions and informatics tasks. His research interests are primarily software architecture and large-scale data-intensive systems. Mattmann received his Ph.D. in computer science from the University of Southern California (USC). Mattmann is an Adjunct Assistant Professor at USC teaching graduate courses in Search Engines and Information Retrieval. Mattmann is also an Executive Officer at the Apache Software Foundation and he is a senior member of IEEE.

**John Harney** joined Oak Ridge National Laboratory in 2010 after earning a Ph.D. in computer science from the University of Georgia. At UGA, he investigated the problem of building Web service compositions using decision-theoretic planners, as well as understanding volatility and its effect on compositions. He has published and presented papers in the WWW, ICSOC and ICWS conferences, and has served as a technical referee in several other venues. At ORNL, he is currently responsible for improving front end portal and search services for the Earth System Grid project, which aims to provide climate scientists a platform to publish and analyze climate data. His current research interests lie in scientific data management, Web services, the semantic Web, and artificial intelligence.

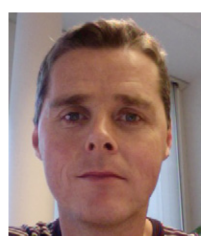
**Galen Shipman** is the Data Systems Architect for the Computing and Computational Sciences Directorate at Oak Ridge National Laboratory. He is responsible for defining and maintaining an overarching strategy for data storage, data management, and data analysis spanning from research and development to integration, deployment and operations for high-performance and data-intensive computing initiatives at ORNL. His research interests span high-performance and data-intensive computing and their application to multiple research domain such as climate science and materials science.

**Feiyi Wang** received his Ph.D. in Computer Engineering from North Carolina State University (NCSU). Before he joined Oak Ridge National Laboratory as research scientist, he worked at Cisco Systems and Microelectronic Center of North Carolina (MCNC) as a lead developer and principal investigator for several DARPA-funded projects. His current research interests include high performance storage system, parallel I/O and file systems, fault tolerance and system simulation, and scientific data management and integration. Dr. Wang is a senior member of IEEE.

**Rachana Ananthakrishnan** is a Sr. Engagement Manager and Solutions Architect at the Computation Institute at University of Chicago and Argonne National Lab. Rachana works in the area of Grid and Cloud Computing, and is a member of the team developing Globus Online, a hosted platform solution with various data management capability for researchers. She has been involved with the Earth System Grid project for number of years, with primary focus on security and data management. She is also currently involved in several other projects, including XSEDE, BIRN and CSSEF. Rachana received her Masters in Computer Science from Indiana University, Bloomington, and a Bachelors in Computer Engineering from Madras University, India.

**Neill Miller** is a senior software engineer at the University of Chicago/Argonne National Laboratory. He has been involved in parallel file system development, monitoring and discovery services for the Grid, and most recently, security, data transfer, and framework elements in the Earth System Grid project.

**Sébastien Denvil**, is a research engineer at CNRS and has a 10 year experience in climate modeling, climate simulations and High Performance Computing. Within the IPSL climate modeling group he is responsible for long IPCC-type simulations and data distribution of model outputs results. He was work package leader during the EU FP7 METAFOR project and he is a work package co-leader of the EU FP7 IS-ENES2 project. He is one of the Principal Investigator of the Earth System Documentation project (ES-DOC) and the G8 ExArch project. He has been IPSL representative in several projects (EU FP6 ENSEMBLES, EU FP7 IS-ENES1, EU FP7 EGI-INSPIRE). He is the national point of contact of the French climate modeling community for data distribution e-infrastructure.



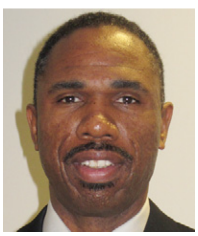
**Mark Morgan** works for the Institut Pierre Simon Laplace (IPSL) in Paris helping to raise the usefulness and maturity of the IPSL's earth system modeling platform. A safe pair of hands & genuine asset to most projects, Mark combines strong design, development & leadership skills with a communicative personality well suited to collaborative teamwork. He is an experienced & motivating team leader (including geographically distributed teams), and currently heads the Earth System Documentation project. Mark is of the view that SOA design underpinned by mature Agile processes reduces risk & promotes creativity.

**Zed Pobre** is a system administrator for the NASA Center for Climate Simulation at the NASA Goddard Space Flight Center, supporting the Discover supercomputing cluster and related data distribution and visualization needs. With over 20 years of software development and systems administration experience, he has also been active in the Free Software community, and is a package maintainer for the Debian GNU/Linux operating system distribution.

**Gavin Bell** is a Senior Computer Scientists/Mathematics Programmer at Lawrence Livermore National Labs in the AIMS group. Gavin has been writing software for over 20 years and he is proud to be helping to coordinate the ESGF.org efforts to build an open-source, community-driven platform to support climate science. His main areas of interest surround building large-scale, distributed, "Big Data" computing platforms, usually involving hundreds of machines operating over petabytes of data. His formal background is in Artificial Intelligence/Machine Learning. Prior to joining LLNL Gavin building trading systems and portfolio management systems on Wall St.

**Charles Doutriaux** is a senior Lawrence Livermore National Laboratory research computer scientist, where he is internationally known for his work in climate analytics, informatics, and management systems supporting model intercomparison projects. He works closely with many world-renowned climate scientists and shares in the recognition of the Intergovernmental Panel on Climate Change 2007 Nobel Peace Prize. His research interests include climate attribution and detection, visualization, and data analysis. Doutriaux has a master's degree in "*Climate and Physico-Chemistry of the Atmosphere*" from the prestigious French University Joseph Fourier, Grenoble. He is a member of the AGU and AMS.

**Bob Drach** works with the Program for Climate Model Diagnosis and Intercomparison, at Lawrence Livermore National Laboratory. He is one of the developers of the Earth System Grid, and helped to build the CMIP3 and CMIP5 multi-model data archives. He is also a principal developer of the Climate Data Analysis Tools. His professional interests include scientific data management and climate data analysis. Bob is a co-author of the CF climate metadata standard.



**Dean N. Williams** is a senior research computer scientist at the Lawrence Livermore National Laboratory, where he is the head of the Analytics, Informatics, and Management Systems program. He is the Principal Investigator for the Earth System Grid Federation and the Ultra-scale Visualization Climate Data Analysis Tools. His research interests are in computer science research, focusing on high-performance and distributed computing, as well as visualization and data analysis. Williams has an MS in computer science from the California State University at Chico. He is a member of IEEE, ACM, and the American Geophysical Union.

**Philip Kershaw** is the Technical Manager for Earth Observation with CEDA, the Centre for Environmental Data Archival at RAL Space, STFC Rutherford Appleton Laboratory in the UK. His specialist areas include federated identity management and cloud computing for the environmental data informatics applications domain. He has been a contributor to the security architecture for the Earth System Grid Federation. He also had a range of experience with Space science and Earth observation related applications development both with STFC and prior to this in industry. His degree was in Electronics and Electrical Engineering.

**Stephen Pascoe** was recently appointed as a Technical Manager within CEDA to acknowledge his expertise with online services and the detailed offline resources needed to support them. Stephen was instrumental in the deployment of a network of services underpinning the high profile UKCP09 site's weather generator service. More recently he has coordinated the technical support from CEDA for the latest CMIP5 distributed data archive, which spans the globe through the Earth System Grid Federation.

**Estanislao Gonzalez** is currently working as a scientific programmer at the Free University of Berlin, Germany, designing an evaluation framework for decadal prediction systems. He previously worked as a data manager at the German Computing Climate Center (DKRZ) in Hamburg, Germany, where he was responsible for the management of the IPCC-CMIP5 archive. He has actively participated in the Earth System Grid Federation and collaborated in many other science related projects. He is focused on problem solving and has a background in informatics, business administration and philosophy.

**Sandro Fiore** received the Ph.D. degree in Computer Science at the University of Lecce in 2004. His research activities focus on HPC, distributed and grid computing, in particular on distributed data management, data analytics/mining and high performance database management. In 2006, he joined the Scientific Computing and Operations Division at CMCC, leading the Scientific Data Management Group. He is a Visiting Scientist at Lawrence Livermore National Laboratory (LLNL) working on the ESGF Dashboard, a proactive distributed monitoring system for the Earth System Grid Federation. He is author of more than 50 papers in refereed books/journals/proceedings on distributed and grid computing.

**Roland Schweitzer** received a Bachelor of Science in Mathematics and French from Oklahoma State University and a Master of Science in Applied Mathematics from Colorado State University. His professional experience began as applications analyst for ETA Systems, the supercomputer subsidiary of the Control Data Corporation. He has since worked his way down the computer scale to the point of developing an award winning weather application for Android smart phones. In between he has accumulated 15 years of experience in all aspects of scientific data management as an employee and independent contractor to universities and NOAA laboratories.