ESGF Meeting / Tech Workshop Notes

FINAL: 2010/07/21

Contents

Attendees

- V. Balaji
- Gavin Bell
- Luca Cinquini
- Bob Drach
- Estanislao Gonzalez
- Stephan Kindermann
- Ross Miller
- Eric Nienhouse
- Serguei Nikonov
- Stephen Pascoe
- F. Wang

Possible project names

- ESGF
- ESGI (Infrastructure)
- COESDI (Community-Owned Earth System Data Infrastructure) *Balaji's favorite*
- CESDI (Community Earth System Data Infrastructure)
- ESDI (Earth System Data Infrastructure)

The term ESG has the advantage that it is a recognised brand but the "G" (Grid) is somewhat devalued these days. Similarly ESGF is recognised as the federation deploying CMIP5 but "Federation" implies people rather than infrastructure and suggests the remit is very close to GO-ESSP. A term replacing "Grid" with "Data" and "Federation" with "Infrastructure" would better fit the remit but would dilute the ESG brand. COESDI is the most descriptive but is a bit long.

Until the name is decided this document will use the name ESGF for the project and ESGF technical working group for the participants at the workshop.

Governance

- The ESGF technical working group operates under the governance of GO-ESSP PIs.
- ESGF's relationship to GO-ESSP: GO-ESSP is a larger group, an open forum for sharing ideas, while ESGF has a much narrower scope, focused on implementing and

operating a working community-owned infrastructure. In the short term, it is focused on enabling CMIP5.
- Central coordination of priorities and gathering of requirements are the responsibility of GO-ESSP PIs
- Technical issues should be delegated by the PIs to the working group. The roadmap below establishes the initial priorities and tasks agreed to by the technical working group.
- ESGF will establish itself in an institution agnostic entity under the project name selected.
- The role of "Coordinator" has been created. This role is given to the individual responsible for managing a given component or set of components (responsibilities described below).

Development Model

- Web Presence and Documentation
    - Decision: evaluate hosting the project on Google Projects, evaluate how much customization can be provided, and what is the functionality (Wiki, documents upload,...)
    - All members of ESGF should have write access to the top-level web pages
    - Top level ESGF page to link back to each component's home site, which should have same look and feel
    - Preferably the top-level domain should look neutral, e.g esgf.org. Alternatively everyone maintains an esgf.my.institution whose top-level homepage looks the same everywhere.
    - project-voldemort.com as an example of multiple projects/components unified under single presentation scheme.
- Licenses
    - All software components must be open source.
    - Recommendation: BSD License is compatible with most open source licenses.
    - Recommendation: Default - Use BSD License used by the esg-node project.
- How to handle tasking and cross-dependence?
    - look at apache commons
    - components with agreed-upon interfaces (needs a process to agree interfaces)
    - first step: access to each others' trackers
    - perhaps a top-level tracker to record cross-dependencies
- Component Management
    - Each system component, or implementation thereof, should have a "Coordinator" that manages its documentation, development, testing and integration with the rest of the system.
    - The component architecture working group, currently the ESGF coordinating members, will define the overall component architecture.
    - Open question: how to manage change to some components that affect other components (example: Metafor live feed breaking the ESG XML parsing)
    - There will be a process such as a new software component is entered into the

official ESGF distribution as either a default or alternative implementation of some functionality.
    - Each component must CLEARLY document its API and strive to maintain its API immutable as the back-end implementation possibly changes.
- Coding Standards
    - Recommendation: for new components, use a scheme to name java packages and python modules.
- Recommended Development Tools
    - Source code repository - GIT
    - Unit Testing - JUnit (Java), Nose (Python)
    - Artifact Repository - Ivy (Java)

ESGF Presentation

- Separate ESGF domain name (.org) and web site.
- Common look and feel for ESGF site and it's components.
- Consistent top *N* page content (overview, about, sponsors, etc...)
- Component Documentation / Web presence: Each component has its own web page with its content hierarchy under the purview of its coordinator and may be hosted at coordinator's home organization.
- Component Versioning: Each coordinator is responsible for the version number of their component(s). Recommended: 4 numeric value for version scheme #.#.#.# (the 4th, least significant value, is optional).
- Progress Reporting: Each component site must have a page expressing the expectations and direction of future development. This may take the form of a Gantt chart, or TODO list with expected completion times.

System Components / Features

The working group identified the components and features that characterise the system. These are itemised below and will form the basis of Information overview and component overview documents (see roadmap)

- Security
    - User registration
    - Authentication (Identity Provider, Relying Party)
    - Attribute Service
    - Authorization Service
- Publisher
    - Production and availability of data catalogs
    - Production and availability of semantic metadata
- Search/Browse
    - Ingestion of data catalogs and semantic metadata
    - Query services
    - Model metadata track-back (inverted index)
- Metadata exchange
- Data Access

    - Files download
        - Browser-based download
        - Scripted bulk download
        - Rich client bulk download
        - Deep storage retrieval
    - Server-side operations
        - Regridding
        - Sub-setting
        - Ensemble statistics
    - Visualization
- Reliable Data Movement
    - Replication
    - Checksums
- Metrics
- Notification
- Monitoring
- Installer scripts
- Framework
    - Inter-component communication (Manager)
    - Introspection (what components are installed)
    - Scheduler

Roadmap

- The group will collaborate on information overview and component overview documents, lead by Stephen.
- The group recommends we extend it's membership to better represent certain key technology areas. Specifically we would like to invite a member in these areas from our other partner institutions.
    - LAS/UI
    - Security
    - Data Movement
    - Metadata
- Gavin to polish the Data Node Manager code such that:
    - It can be bootstrapped without any component dependency
    - Does not require any other peer by default (peers can be registered afterward)
    - A sample "Echo" component can demonstrate invocation of events and retrieval of information
    - Gavin will provide design document on Data Node manager framework
- Luca to write XML parsing code that allows for configuration-driven registration of components
- Gavin to setup tools environment that includes Git, Ivy, Junit and Hudson (with some little documentation)
- Bob: determine if PCMDI can provide documentation web site and support at PCMDI using Plone, with neutral domain name like esgf.org. For now, the site will contain a set of links to external documentation that must be open. Later, revisit how the web site is

working and possibly move everything to another CMS.

- Luca to insert the security filters and ORP in the GIT repository
- Luca to work on solr-based search:
    - Insert into datanode manager framework
        - Need access to PCMDI GIT repository
    - Possibly insert into current Gateway framework
        - Includes plugins for faceted search on the gateway
        - Includes metadata exchange via OAI
        - Need access to NCAR SVN repository
- ORP filter authentication will be an option for datanodes by the August 16th release of the Gateway. Datanodes can decide which authorisation filter to use and Gateways will be able to support either. Gateways at PCMDI, NCAR and BADC will be updated to support this feature.
- ORNL to experiment with providing a data node component that generates wget scripts once it receives a set of dataset ids and possibly a set of filters like time range, variable name match, product=request|output etc.
- DKRZ to generalize the staging filter to any mass store and make it available as part of the ESG distribution
- Search on complex metadata: start conversation with both Metafor and ESC teams to possibly provide a component that performs search and browsing of complex metadata objects:
    - Modularization of ESC trackback functionality
    - Search on CIM XML documents
- Evaluate how Python modules can be invoked from the Data Node manager framework
    - Luca, Bob, Stephen, ...: Wrap python calls via a Python application framework (pylons, django, webpy, etc.) to localhost (same as CGI scripting)
    - Or, use Apache in front of Tomcat and python engine
- Establish monthly teleconference of the ESGF Technical Working Group - first one around end of August