ApacheCon North America

Portland, Oregon 26th – 28th February 2013

Leveraging Tools and Components from OODT and Apache within Climate Science and the Earth System Grid Federation

Luca Cinquini, Dan Crichton, Chris Mattmann

NASA Jet Propulsion Laboratory,
California Institute of Technology

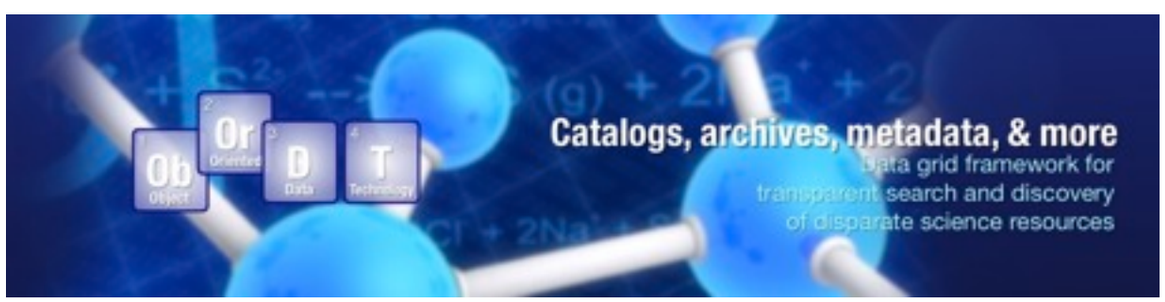
Outline

This talk will present the Earth System Grid Federation (ESGF) as an example of a scientific project that is leveraging tools from the ASF to successfully manage and distributed large amounts of scientific data.

- The Earth System Grid Federation (ESGF)
  - Science motivations
  - High level architecture
  - Major software components: Node Manager, Search Service, Web Front End

- Apache Solr
  - General functionality
  - Scalability options
  - Usage within ESGF

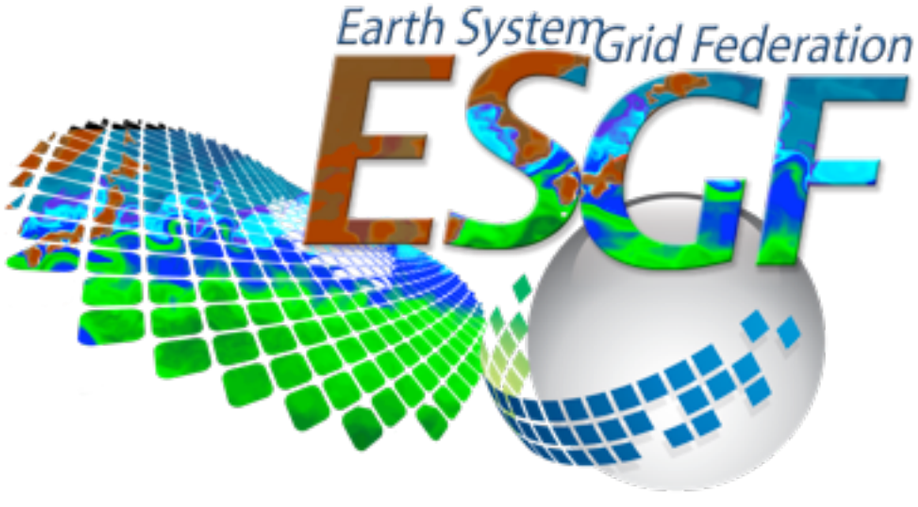
- Apache Object Oriented Data Technology (OODT)
  - Description of OODT components
  - Current usage for ESGF publishing
  - Next generation ESGF data processing



Introduction

The Earth System Grid Federation (ESGF) is a multi-agency, international collaboration of people and institutions working together to build an open source software infrastructure for the management and analysis of Earth Science data on a global scale

- Historically started with DoE Earth System Grid project, expanded to include groups and institutions around the world that are involved in management of climate data
- Recently evolved into a next generation architecture: the Peer-To-Peer (P2P) system (higher performance, more reliable and scalable, focused on federation services)



Acknowledgments

- Software development and project management: ANL, ANU, BADC, CMCC, DKRZ, ESRL, GFDL, GSFC, JPL, IPSL, ORNL, LLNL (lead), PMEL, ...
- Operations: tens of data centers across Asia, Australia, Europe and North America
- Funding: DoE, NASA, NOAA, NSF, IS-ENES,...

Support for Climate Change Research

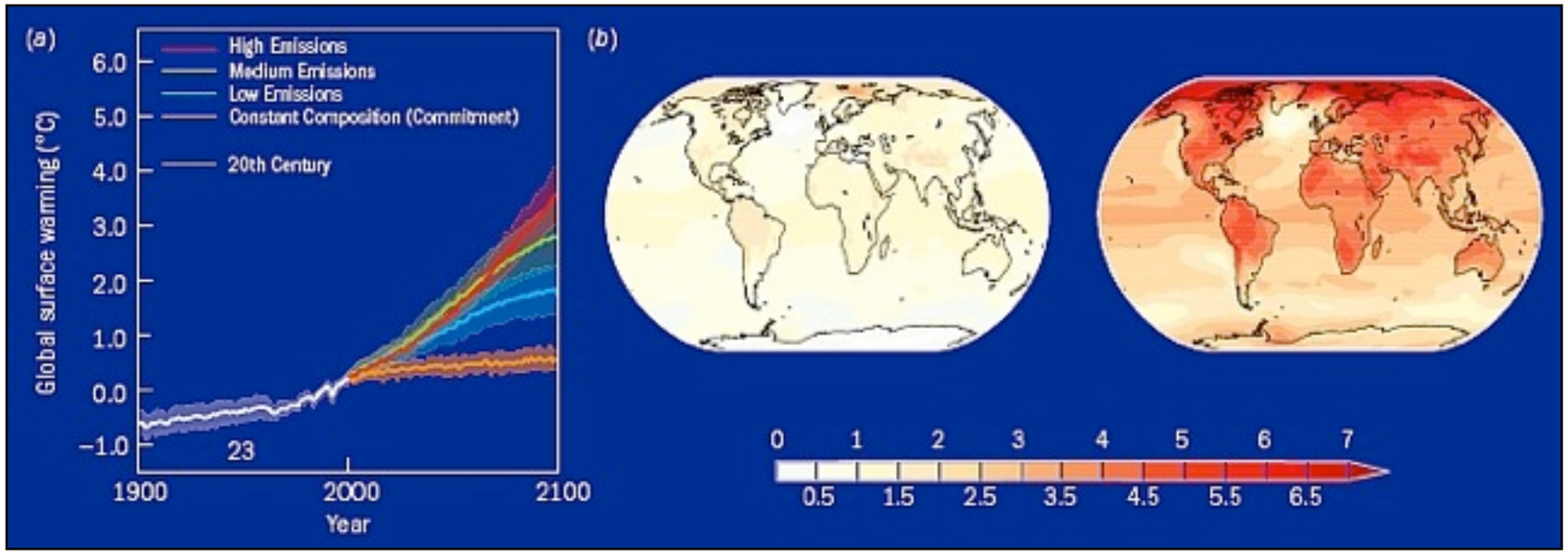
- ESGF primary goal is to support research on Climate Change - one of the most critical and complex scientific problems of our time
- IPCC (International Panel on Climate Change): international body of scientists responsible for compiling reports on Climate Change to the global community and political leaders
- Over the years, IPCC reports have increasingly supported evidence for climate change and identified human causes for this change

- Next IPCC-AR5 report due in 2014, will be based on data produced by CMIP5 modeling project (and delivered by ESGF)

CMIP5: a “Big Data” problem

CMIP5: Coupled Model Intercomparison Project - phase 5

- Arguably the largest coordinated modeling effort in human history
- Coordinated by WCRP (World Climate Research Program)
- Includes 40+ climate models in 20+ countries
  - Models are run to simulate future state of the Earth under different emission scenarios
  - Common set of physical fields (temperature, humidity, precipitation etc.) are generated
  - Predictions from different models are compared, averaged, scored



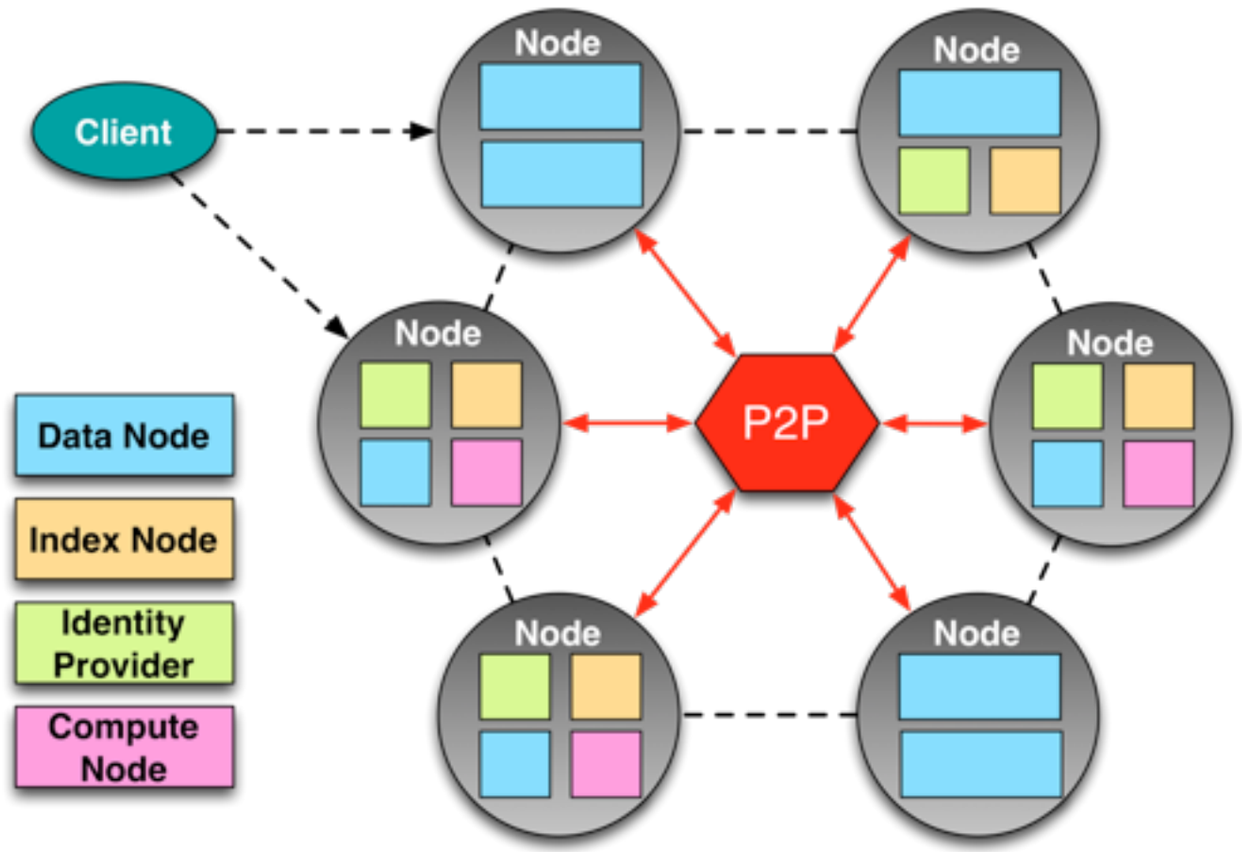
“Big Data” characteristics

- Volume: approx. 2.5 PB of data distributed around the world
- Variety: must combine model output with observations of different kind (satellite, in-situ, radar, etc.) and reanalysis
- Veracity: data and algorithms must be tracked, validated
- Velocity: N/A as immediate availability of model output is *not* important

System Architecture

ESGF is a system of distributed and federated Nodes
that interact dynamically through a Peer-To-Peer (P2P) paradigm

- Distributed: data and metadata are published, stored and served from multiple centers ("Nodes")
- Federated: Nodes interoperate because of the adoption of common services, protocols and APIs, and establishment of mutual trust relationships
- Dynamic: Nodes can join/leave the federation dynamically - global data and services will change accordingly

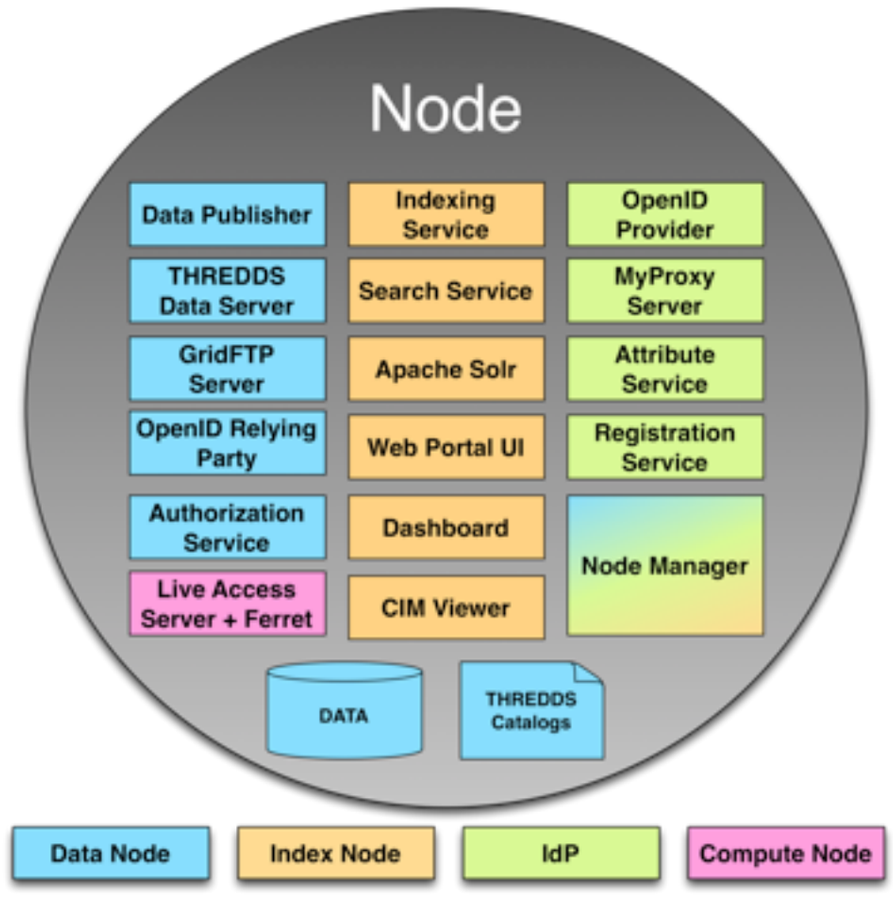


A client (browser or program) can start from any Node in the federation and discover, download and analyze data from multiple locations as if they were stored in a single central archive.

Node Architecture

- Internally, each ESGF Node is composed of services and applications that collectively enable data and metadata access, and user management.
- Software components are grouped into 4 areas of functionality (aka “flavors”):

- Data Node : secure data publication and access
- Index Node :
  - metadata indexing and searching (w/ Solr)
  - web portal UI to drive human interaction
  - dashboard suite of admin applications
  - model metadata viewer plugin
- Identity Provider : user registration, authentication and group membership
- Compute Node : analysis and visualization



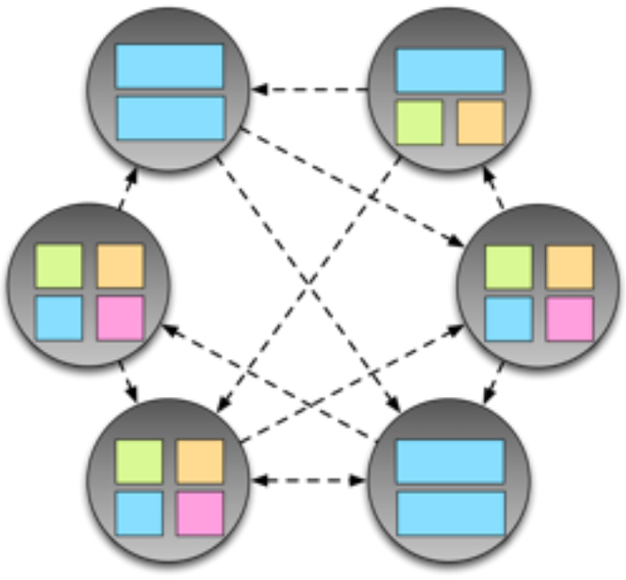
- Nodes flavors can be installed in various combinations depending on site needs, or to achieve higher performance and scalability
- Node Manager is installed for *all* Node flavors to exchange service, state information

The Node Manager and P2P Protocol

- Enables continuos exchange of service and state information among Nodes
- Internally, it collects Node health information and metrics (cpu, disk usage, etc.)

Peer-To-Peer (P2P) protocol

- Gossip protocol: information is exchanged randomly among peers
  - Each Node receives information from one Node, merges it with its own information, and propagates it to two other Nodes at random
  - No central coordination, no single point of failure
- Nodes can join/leave the federation dynamically
- Each Node is bootstrapped with knowledge of one default peer
- Each Node can belong to one or more peer groups within which information is exchanged



XML Registry

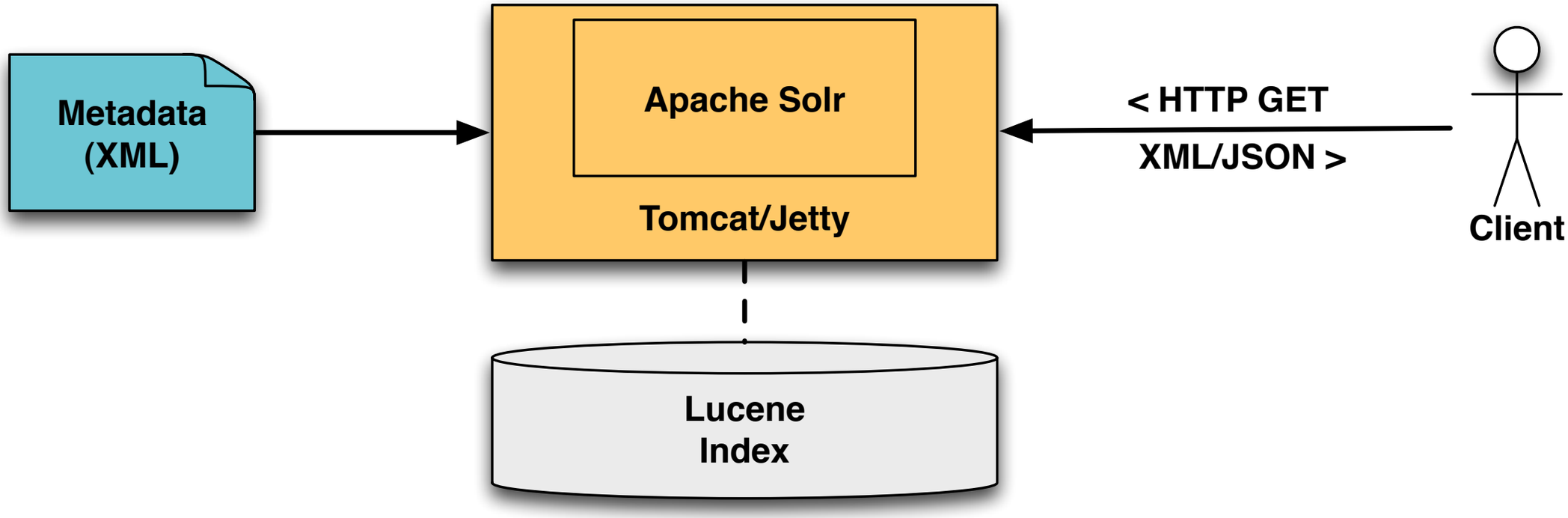
- XML document that is payload of P2P protocol
- Contains service endpoints and SSL public keys for all Nodes in the federation
- Derived products (list of search shards, trusted IdPs, location of Attribute Services,...) are used by federation-wide services

Challenge: *good* news travel fast, *bad* news travel slow...

Apache Solr

Solr is a high-performance web-enabled search engine built on top of Lucene

- Used in many e-commerce web sites
- Free text searches (w/ stemming, stop words, ...)
- Faceted searches (w/ facet counts)
- Other features: scoring, highlighting, word completion,...
- Generic flat metadata model: (key, value+) pairs

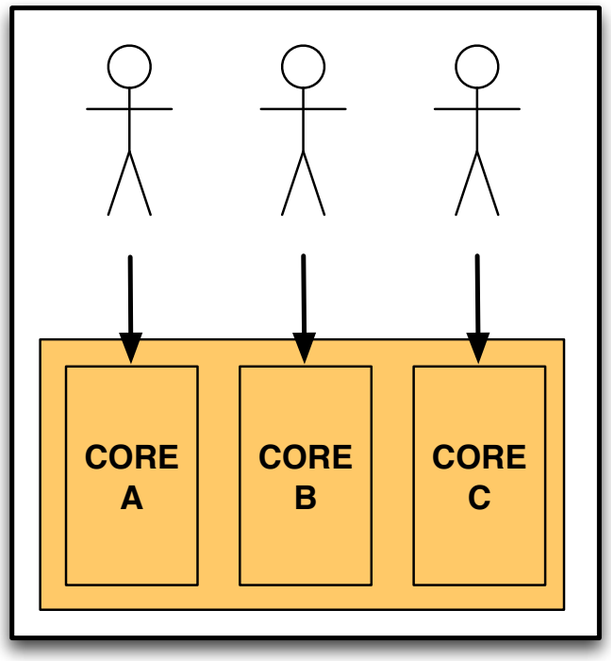


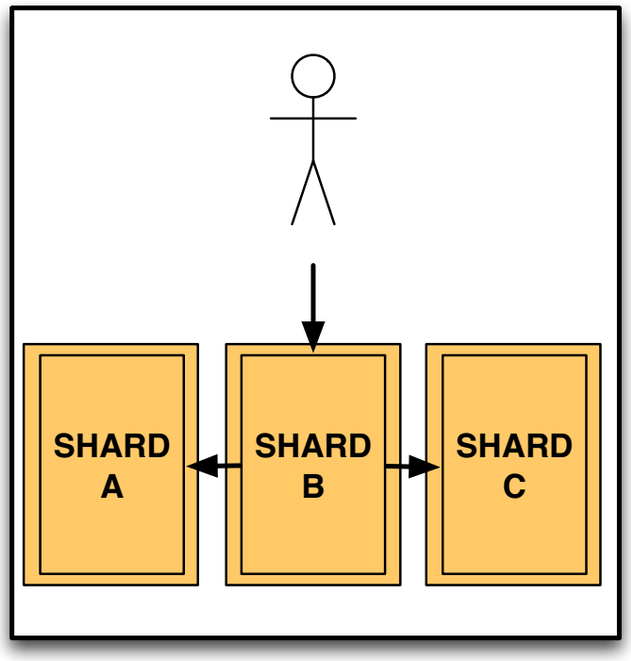
Because of it high performance and domain-agnostic metadata model, Solr is well suited to fulfill the indexing and searching needs of most data-intensive science applications.

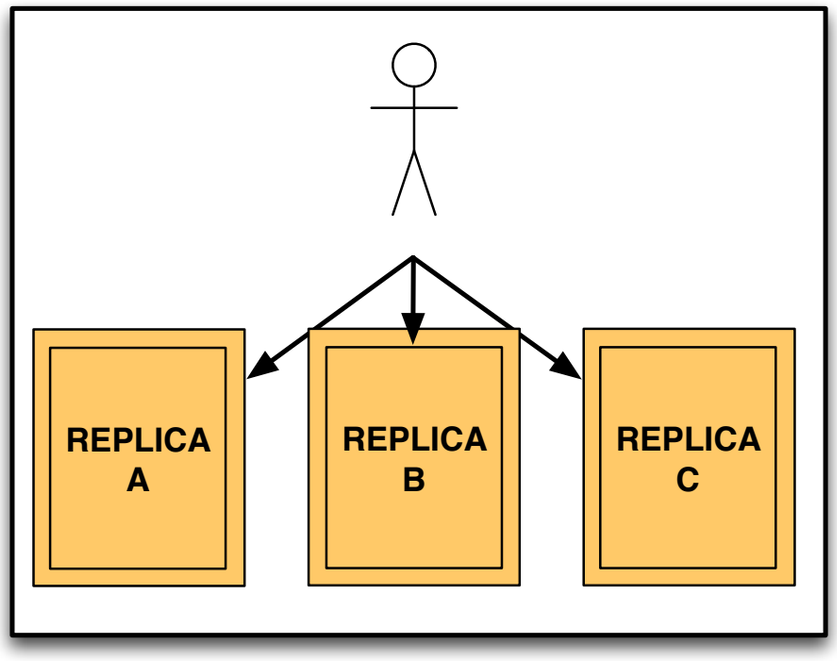
Solr Scalability

Solr has advanced features that allow it to scale to tens of M of records:

- Cores: separate indexes containing different record types that must be queried independently. Cores run within same Solr instance.
- Shards: separate indexes containing records of the same type: query is distributed across shards, results merged together. Shards run on separate Solr instances/servers.
- Replicas: identical copies of the same index, for redundancy and load balance. Replicas run on separate Solr instances/servers.



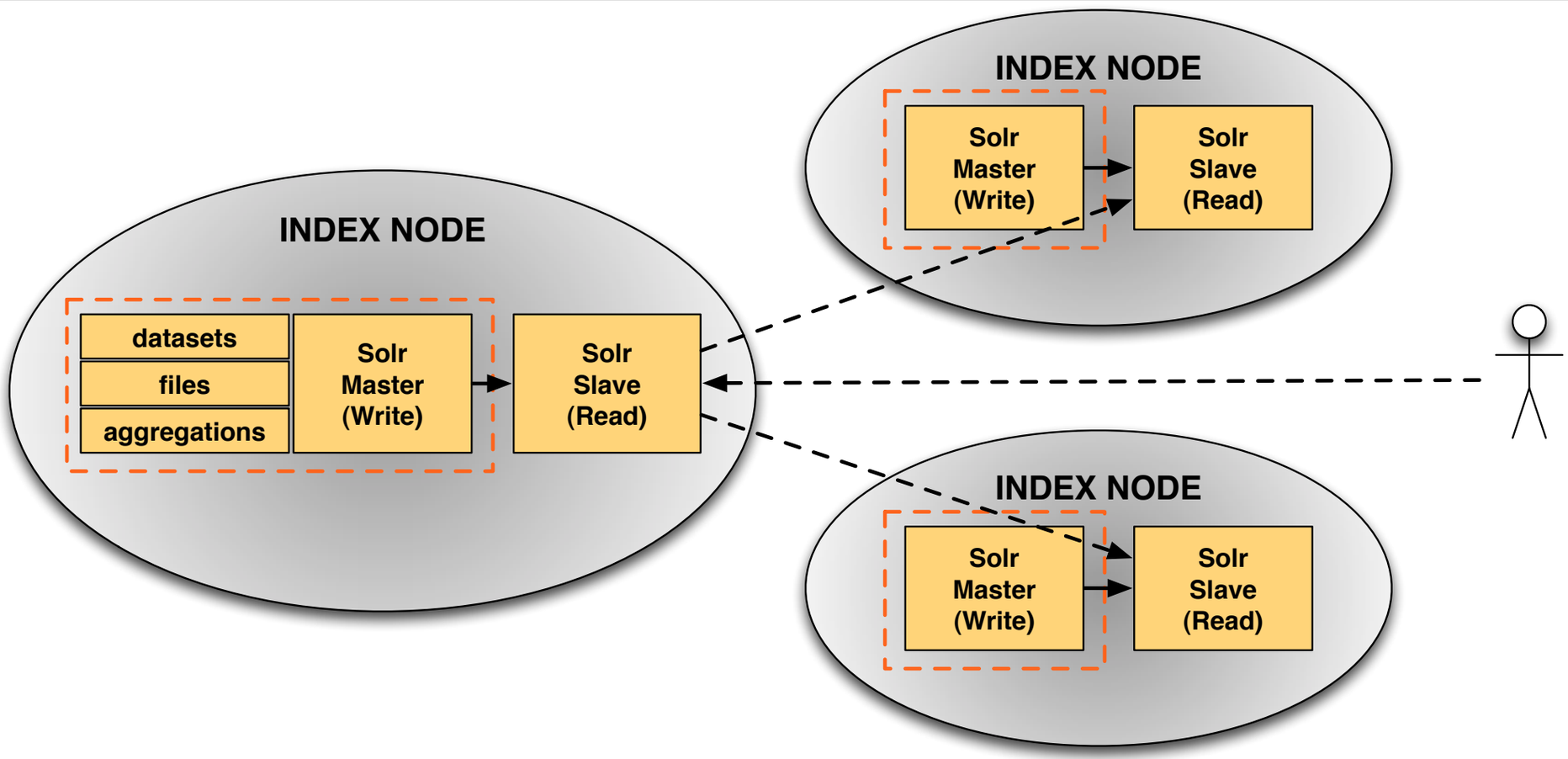




ESGF utilizes all the above techniques to attain satisfactory performance over large, distributed data holdings

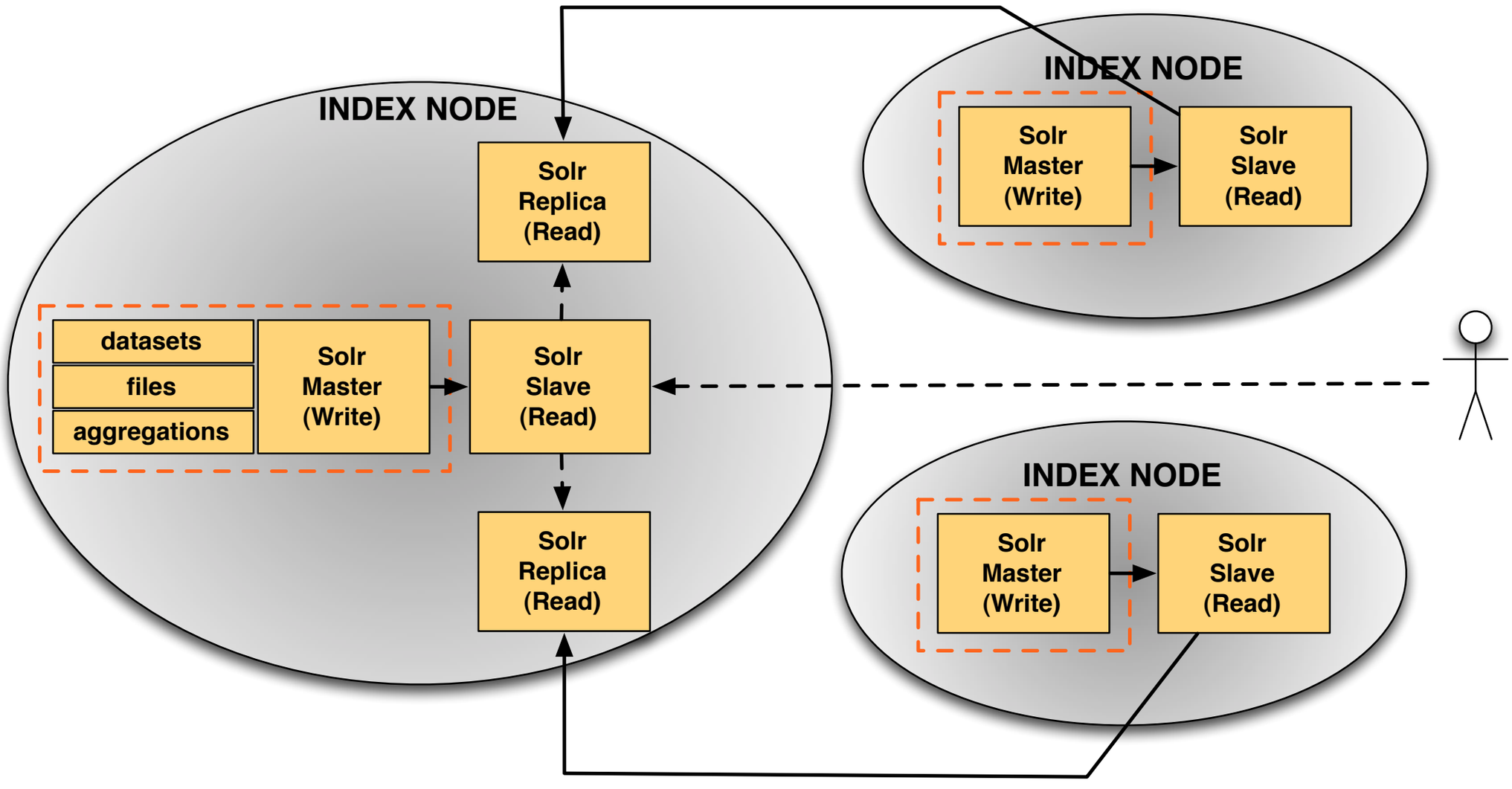
ESGF Search Architecture

- Each Index Node runs a Master Solr (:8984) and a Slave Solr (:8983)
  - Metadata is written to the Master Solr, replicated to the Slave Solr
  - Publishing operations ("write") do not affect user queries ("read")
  - Master Solr is secured within firewall, enabled for writing (and reading)
  - Slave Solr is open to the world for reading but *not* enabled for writing
- A query initiated at any Slave Solr is distributed to all other Slave Solrs in the federation
- Internally, each Master/Slave Solr runs multiple cores:
  - "datasets" core contains high level descriptive metadata for search and discovery
  - "files", "aggregations" cores contain inventory-level metadata for data retrieval



ESGF Search Architecture

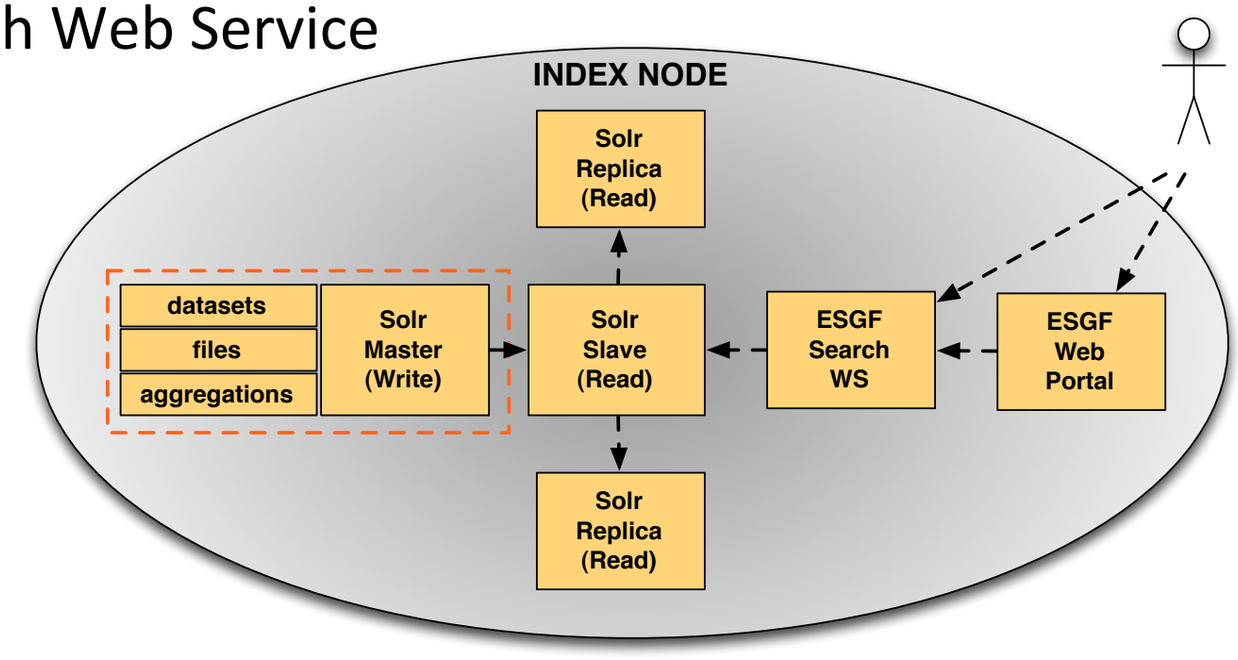
- Each Index Node may create local replicas of one or more remote Solr Slaves
  - Distributed query operates on local indexes only - no network latency!
  - Local query gives complete results even if remote Index Node is unavailable
  - Additional resources are needed to run multiple Slave Solrs at one institution



ESGF Search Web Service

ESGF clients do not query Solr directly, rather they query the ESGF Search Web Service that front-ends the Solr slave:

- Exposes a RESTful query API simpler than Solr query syntax
  - Query by free text
  - Query by domain-specific facets: /search?model=...&experiment=...&time_frequency=...
  - Extremely popular with client developers
- Provides higher-level functionality:
  - automatic query distribution to all known shards (dynamically!)
  - pruning of unresponsive shards
  - generation of wget scripts with same query syntax: /wget?model=...&experiment=...
  - generation of RSS feeds for latest data across the federation
- Insulates client versus possible (unlikely) future changes of back-end engine technology
- ESGF Web Portal is a client to the ESGF Search Web Service



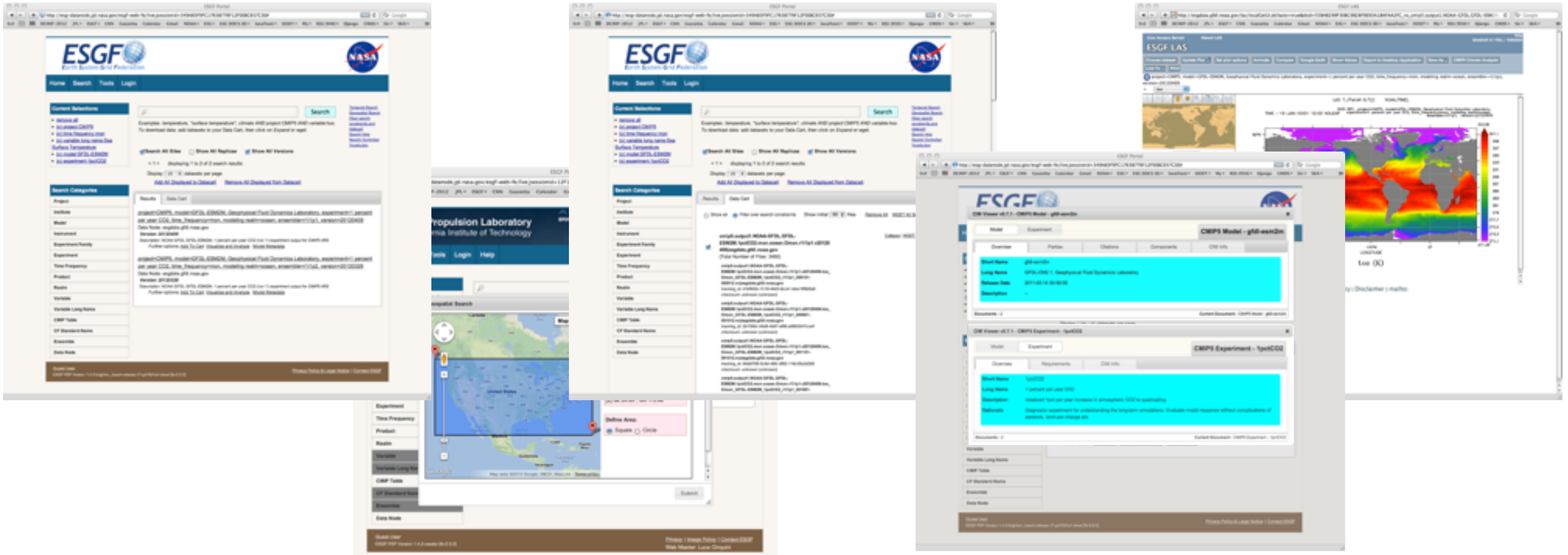
Web Portal Front End

ESGF Web Portal: web application to allow human users to interact with the system

- User registration, authentication and group membership
- Data search, access, analysis and visualization

The Web Portal Search UI allows users to formulate search requests and access data:

- Search by free text, facets, geo-spatial and temporal constraints
- Datasets added to “data cart”
- Files downloaded via single HTTP links, wget scripts, Globus Online (experimental)
- Hyperlinks to higher services for data analysis and visualization and metadata inspection



ESGF Global Distributed Search

ApacheCon North America
Portland, Oregon 25th - 28th February 2013

- Query from any ESGF web portal is distributed to all other web portals
- Users can query in real time a world-wide database containing tens of M of records

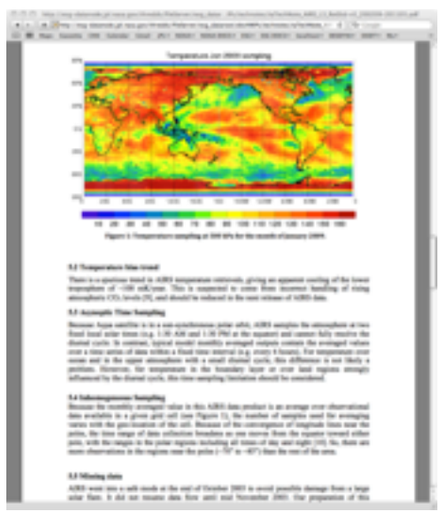
Query Pattern

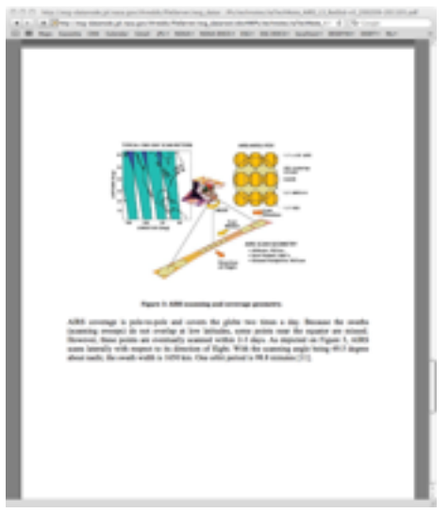
- "Dataset"-level query is distributed to multiple Index Nodes
- "File"-level query issued versus a single specific Index Node

NASA Obs4MIPs Datasets

- ESGF allows access not only to model output, but also observational and reanalysis datasets
- NASA Obs4MIPs: selected satellite observations especially prepared for usage in IPCC-AR5, formatted as output from global climate models:
  - Formatted as NetCDF/CF
  - Level 3 products: global lat/lon grids (not satellite swaths)
  - CMIP5 metadata conventions for search and discovery
  - Include detailed Technical Notes to instruct on proper use of observations
- Goal: facilitate comparison of observations with models and therefore quantify the reliability of model predictions for climate change ("scoring")
- Preparation and publication of Obs4MIPs datasets into ESGF was accomplished by establishing a data processing pipeline that leveraged OODT components (CAS File Manager and CAS Curator)


Atmospheric Infrared Sounder/Advance Microwave Sounding Unit (AIRS/AMSU)
Air Temperature Description





Apache OODT

OODT: Object Oriented Data Technology: framework for management, discovery and access of distributed data resources. Main features:

- Modularity: framework of standalone components that can be deployed in various configurations to fulfill a project specific requirements
- Configurability: each component can be easily configured to invoke alternate out-of-the-box functionality or deployment options
- Extensibility: components can be extended by providing alternate implementations to its core APIs (expressed as Java interfaces) or configuring custom plugins

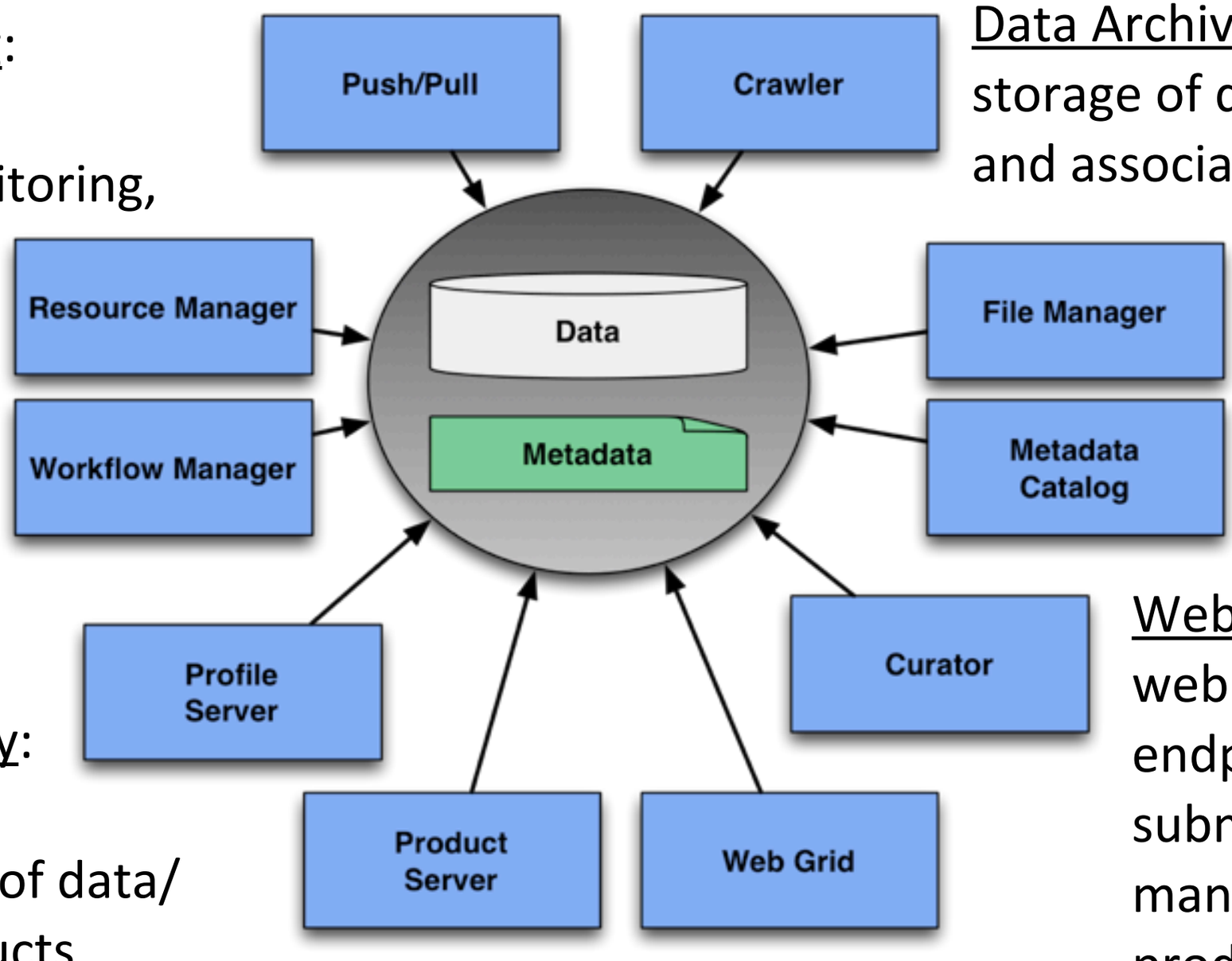
Apache OODT home: http://oodt.apache.org/
Apache OODT wiki: https://cwiki.apache.org/confluence/display/OODT/Home
Apache OODT Jira: https://issues.apache.org/jira/browse/OODT

Apache OODT Components

OODT is an “eco-system” of “data/metadata-centric” software components that altogether provide extensive functionality for science data management needs

Data Movement: push or pull data from remote locations, crawling staging areas

Data Processing: job submission, execution, monitoring, coordination



Data Archiving: storage of data resources and associated metadata

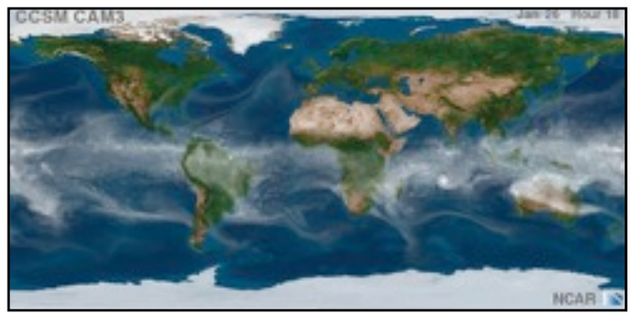
Product Delivery: retrieval and transformation of data/ metadata products

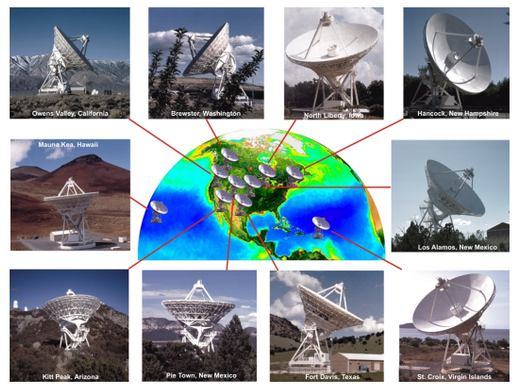
Web Applications: web UI and RESTful endpoints for job submission, metadata management, product access

Apache OODT Adoption

OODT is used operationally to manage scientific data by several projects in disparate scientific domains:

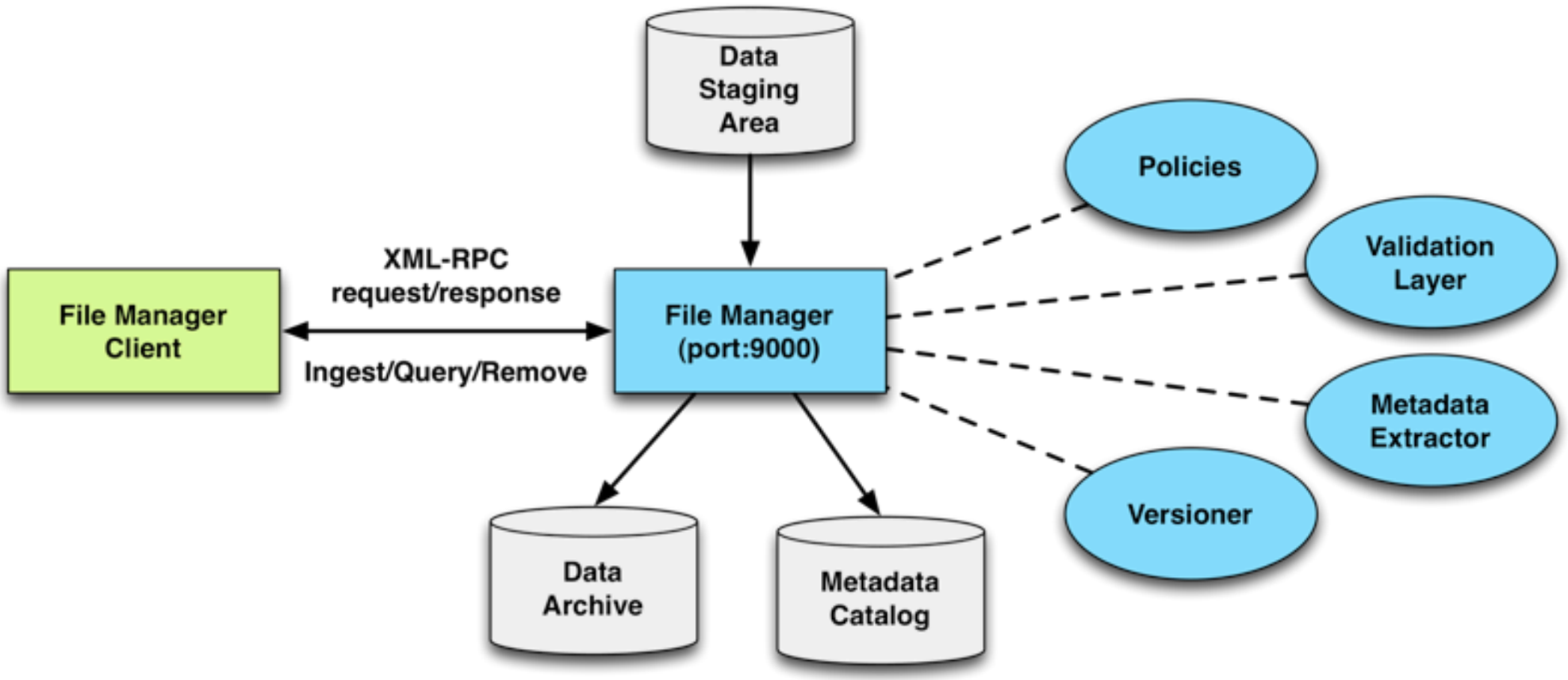
- Earth Sciences:
  - NASA satellite missions (SMAP,...) are using OODT components as the base for their data processing pipeline for generation and archiving of products from raw observations
  - ESGF (Earth System Grid Federation) used OODT to build and publish observational data products in support of climate change research
- Health Sciences: EDRN (Early Detection Research Network) uses OODT to collect, tag and distribute data products to support research in early cancer detection
- Planetary Science: PDS (Planetary Data System) is developing data transformation and delivery services based on OODT as part of its world-wide product access infrastructure
- Radio Astronomy: several projects (VLBA, ALMA, Haystack,...) adopting OODT based on successful VASTR example





OODT CAS File Manager

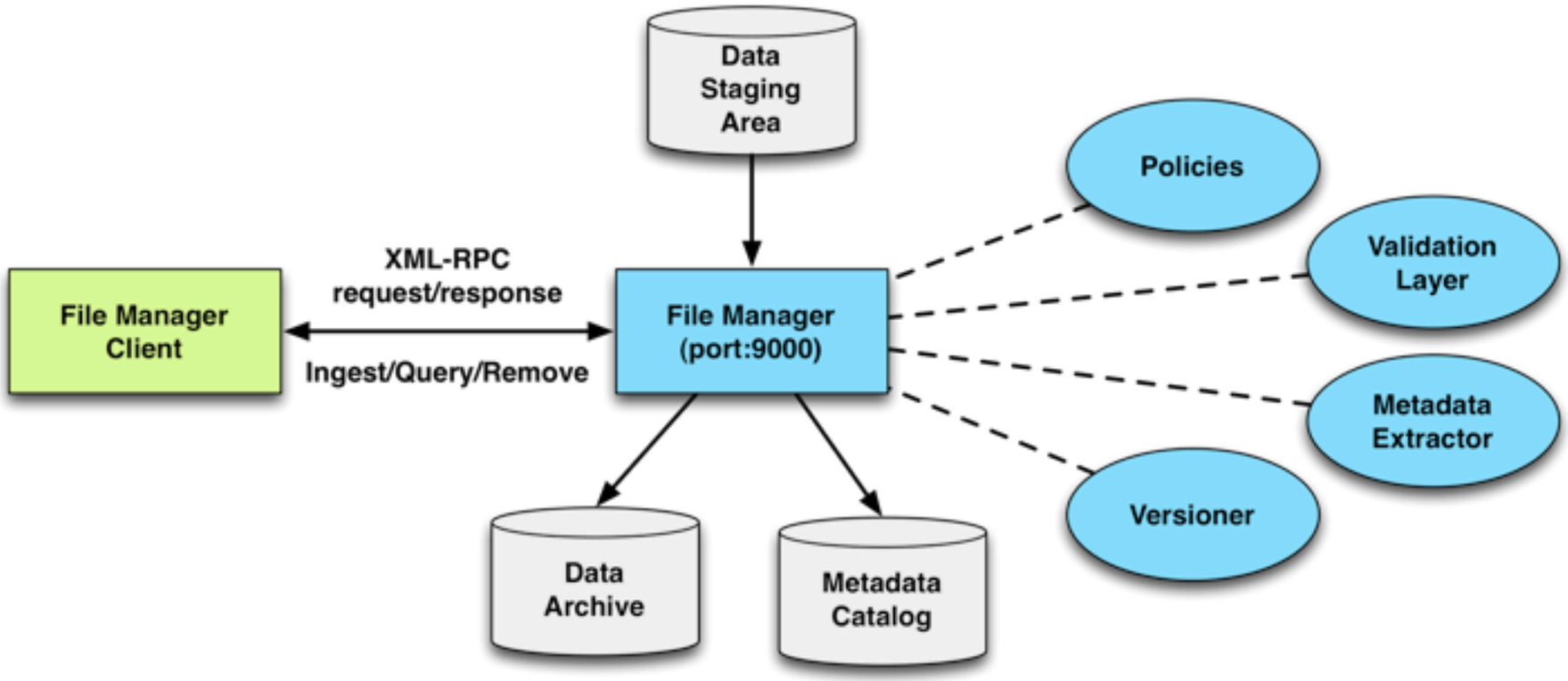
- Purpose: service for cataloging, archiving and delivery of data products (files and directories) and associated metadata
- Example Use Case: science mission or climate model that generates files in a staging area. Files are submitted to the File Manager to:
  - Extract/generate project specific metadata
  - Move files to final archive location
  - Store metadata and file location references into metadata store
  - Later, clients may query product metadata and retrieve data products via XML-RPC



OODT CAS File Manager

The CAS File Manager provides many configuration opportunities ("extension points") that allow customization to project specific needs:

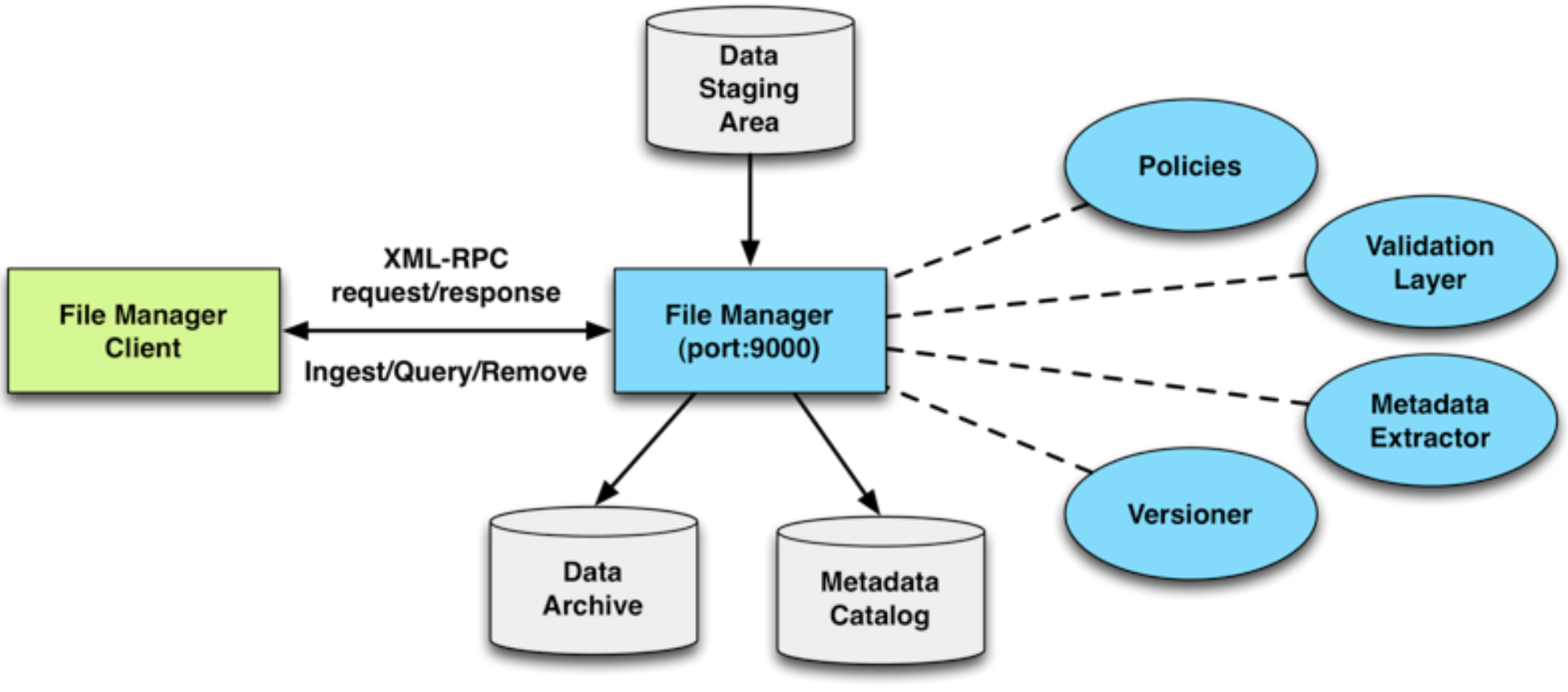
- Policy Files: XML schema instances that specify the supported Product Types and their associated metadata elements
  - Can ingest files of any type: music, videos, images, binary files (e.g. NetCDF), etc.
  - Products can be *Flat* (all files in one dir) or *Hierarchical* (span multiple dirs)
- Metadata Extractors: each ProductType can be associated with a specific Metadata Extractor which is run when the product is ingested
  - User can write their own metadata extractor in Java...
  - ...or use the *ExternMetExtractor* to run a metadata extractor in any other language



Note: check out Apache Tika for flexible metadata extraction

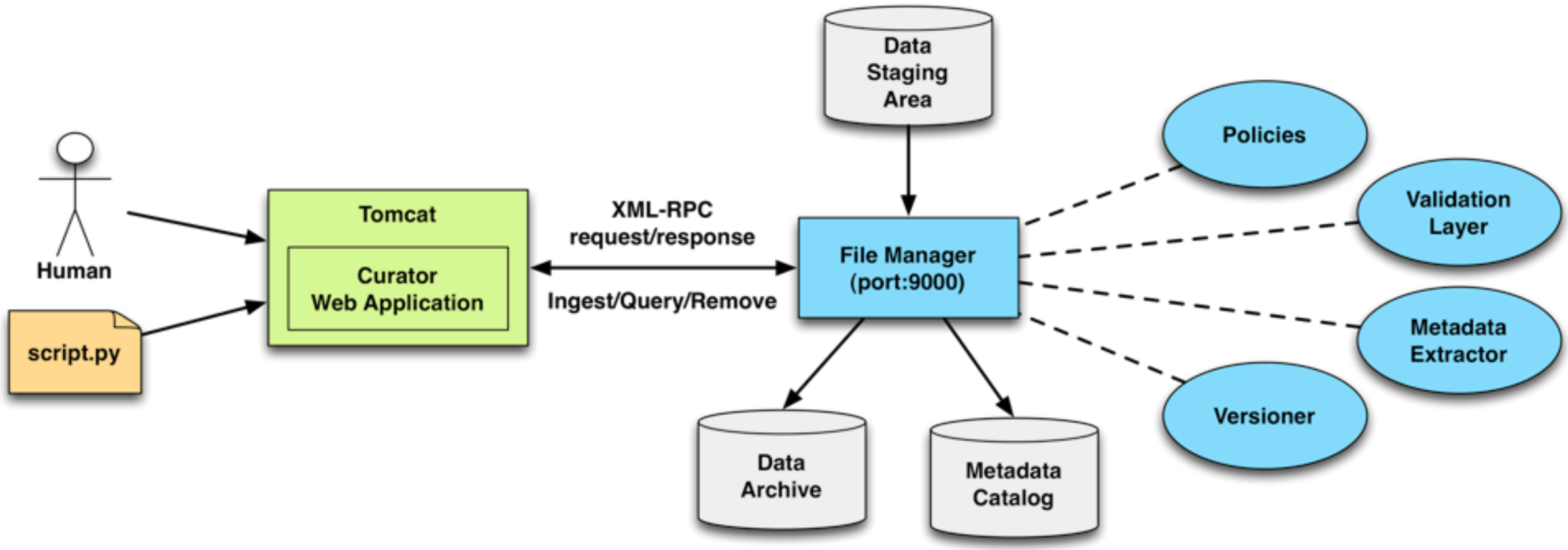
OODT CAS File Manager

- Metadata Catalog: extracted metadata can be stored in various back-end stores
  - Out-of-the-box implementations for Lucene, MySQL and Oracle
  - Solr back-end implementation forthcoming
- Data Transfer Protocol: files may be retrieved from the local file system or over HTTP, stored in-place or moved to a different target destination for archiving
- Validation Layer: inspects products and metadata before ingestion for required metadata elements
- Versioner: specifies how data products are organized within archive



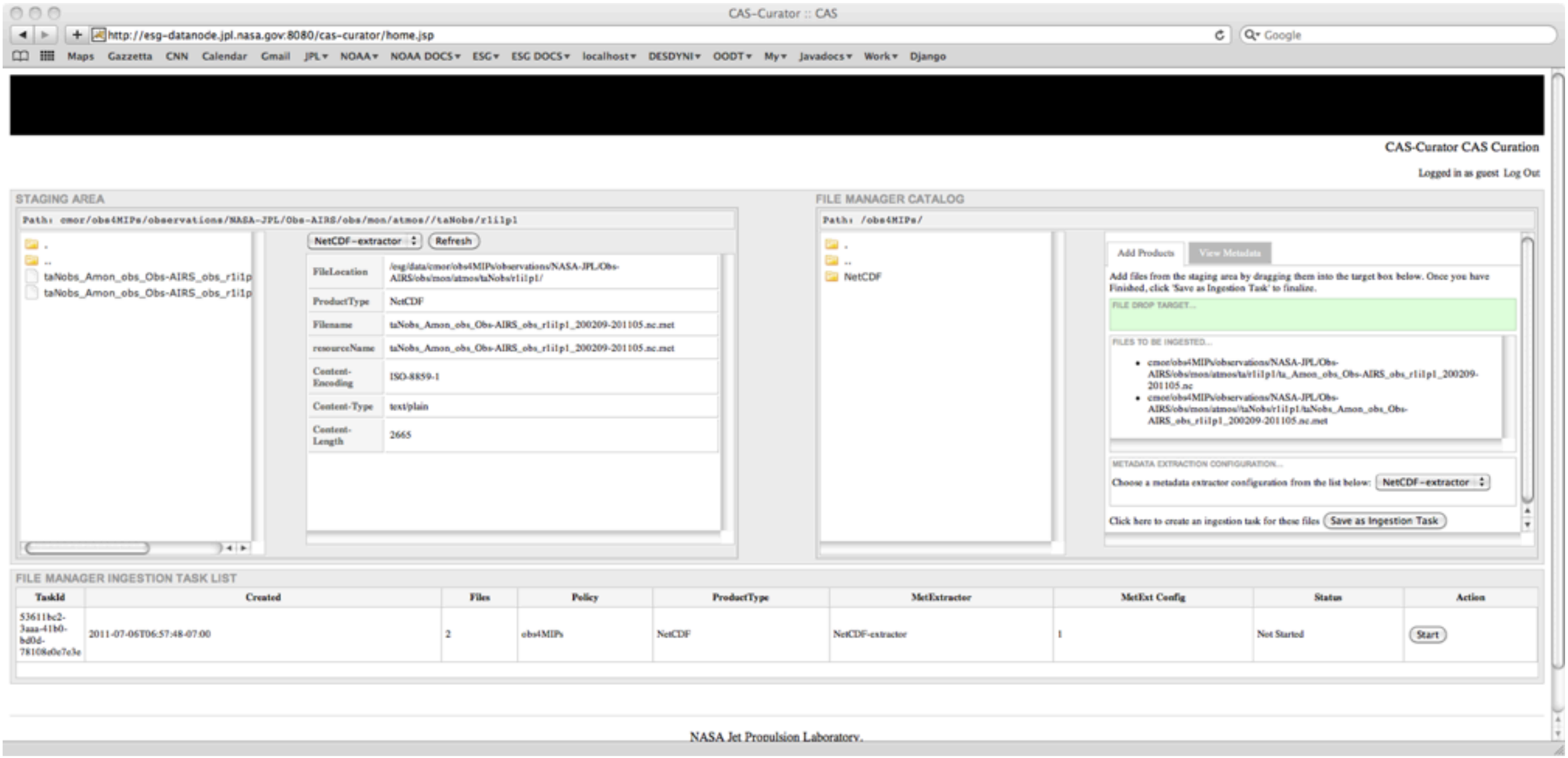
OODT CAS Curator

- Purpose: web application for interacting with the File Manager (i.e. web-based client to the File Manager service):
  - Submit data ingestion jobs
  - Generate, inspect, update and add metadata ("curation")
- Features:
  - Web user interface (human-driven interaction) + REST API (machine-machine interaction)
  - Runs within generic servlet container (Tomcat, Jetty, etc.)
  - Pluggable security layer for authentication and authorization



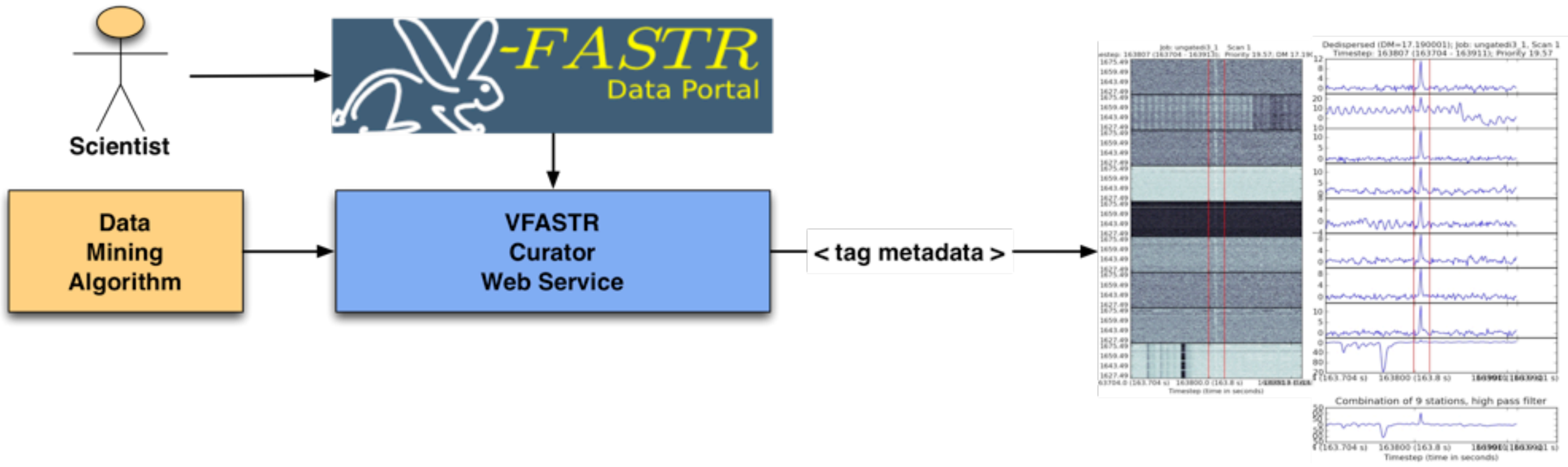
OODT CAS Curator

- Web User Interface: used by humans to manually interact with the system
  - Drag-and-drop selection of files from staging area
  - Selection of metadata extractor from available pool
  - Submission of job for bulk ingestion to File Manager



OODT CAS Curator

- Web REST API: used by programs and scripts for automatic interaction
  - Based on Apache JAX-RS project (project for Restful web services)
  - Retrieve, list and start ingestion tasks
  - Annotate existing products with enhanced metadata
- Example HTTP/POST invocation:
  - curl --data “id=<product_id>&metadata.<name>=<value>” http://<hostname>/curator/services/metadata/update

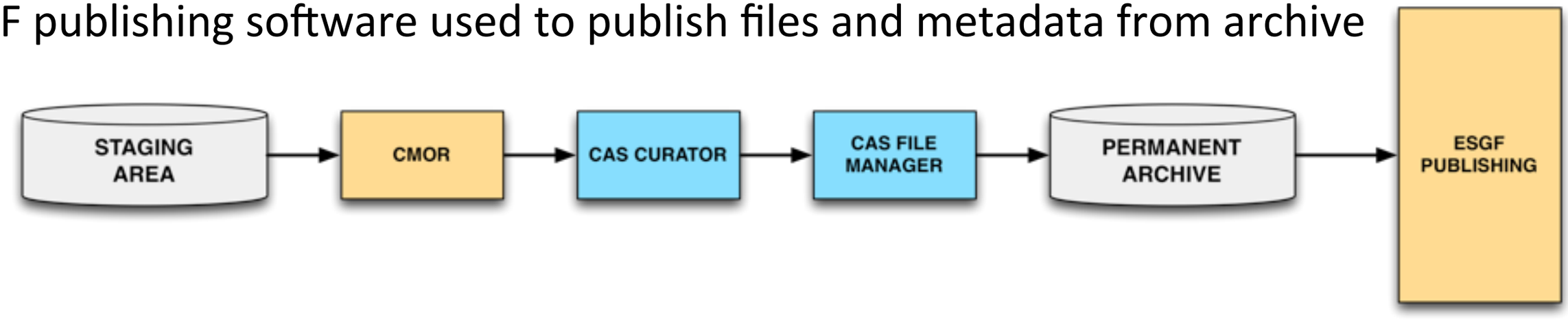


Obs4MIPs Publishing Pipeline for IPCC-AR5

ApacheCon North America
Portland, Oregon 25th - 28th February 2013

OODT File Manager and Curator components were used to setup a data processing pipeline to publish Obs4MIPS datasets into ESGF for use by IPCC-AR5 climate studies

- Obs4MIPs datasets were prepared by NASA mission+data teams and transferred to a staging directory at JPL
- Files were post-processed by CMOR
  - CMOR: climate package to add required metadata such as NetCDF global attributes
- Curator web interface was used to submit ingestion jobs to the File Manager
  - Custom Versioner used to re-organize files according to CMIP5 directory structure: /<project>/<institution>/<time_frequency>/<variable>/..../<filename>
  - Custom Obs4MIPs policies defined the required metadata elements
  - Custom metadata extractor used to parse file content and directory path to generate metadata content
  - Validation Layer enforced existence of required metadata elements
- ESGF publishing software used to publish files and metadata from archive



CMAC "next generation" Obs4MIPs

- Current Obs4MIPs publishing infrastructure has several limitations:
  - Manual: creation of Obs4MIPs datasets was a very labor intensive process executed by DAAC teams
  - Not generic: each DAAC developed a one-off software solution not conforming to any specification or API
  - Not scalable: publication of Obs4MIPs datasets was accomplished manually via Curator web UI

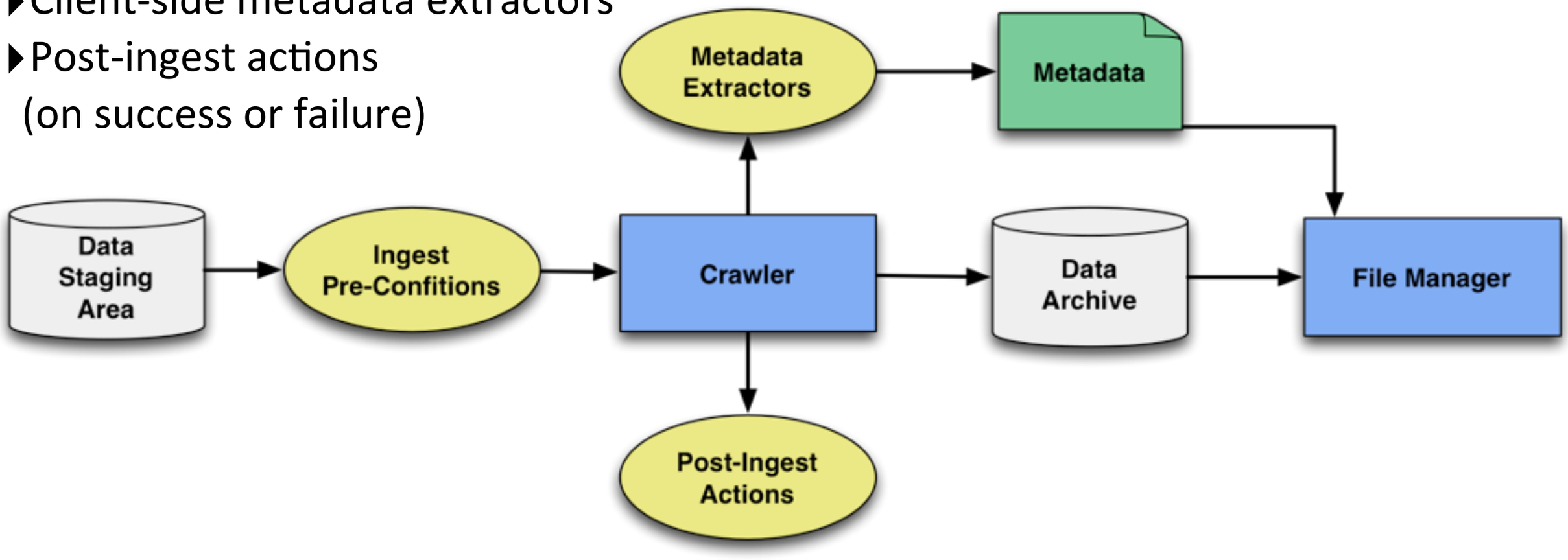
- NASA centers (JPL, GSFC, LaRC) are working on a next generation Obs4MIPs infrastructure that will facilitate preparation and publication of Obs4MIPs datasets and comparison with model output. Goals:
  - Generalize the preparation of Obs4MIPs datasets by defining an API that can be implemented by each DAAC for each specific observing mission
  - Provide tools for validation of generated products
  - Automatically publish product to ESGF
- Funding: NASA CMAC (Computational Modeling Algorithms and Cyber-infrastructure)
- Looking at building an automatic Obs4MIPs data processing and publication pipeline, to be installed at each DAAC, using OODT components: Crawler and Workflow Manager

CAS Crawler

The CAS Crawler is an OODT component that can be used to list the contents of a staging area and submit products for ingestion to the CAS File manager. Typically used for automatic detection of new products transferred from a remote source.

Features:

- Runs as single scan or as time-interval daemon
- Wide range of configuration options:
  - Ingestion pre-conditions
  - Client-side data transfer
  - Client-side metadata extractors
  - Post-ingest actions
    (on success or failure)

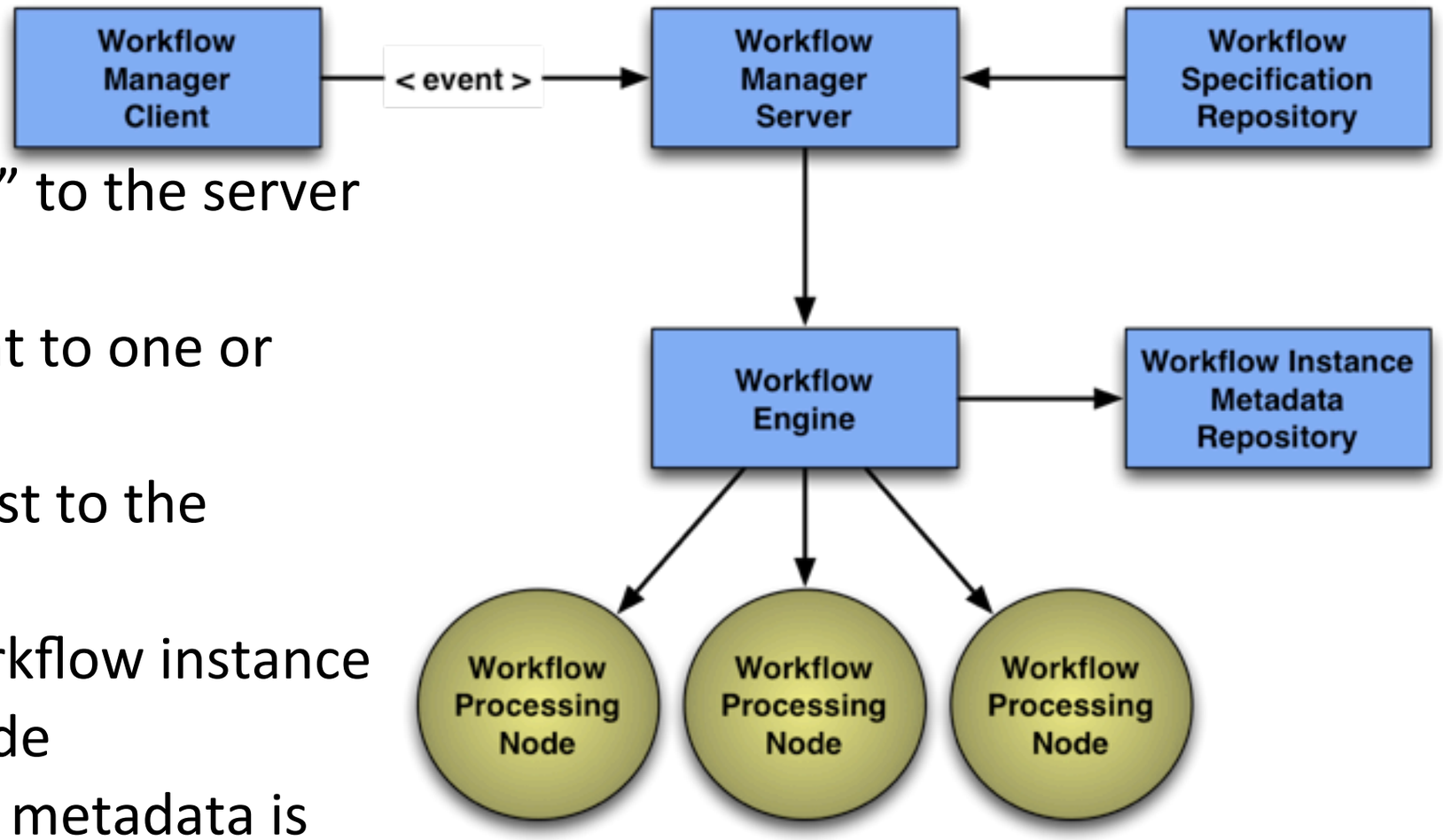


CAS Workflow Manager

The CAS Workflow Manager (WM) is an OODT component for the definition, execution and monitoring of workflows. A workflow is an ordered sequence of processing tasks, where each task reads/processes/writes data. Typically used to establish processing pipelines for scientific data, for example data streaming from NASA observing systems.

<u>Client-Server Architecture</u>

- WM client submits an "event" to the server via XML/RPC protocol
- WM server matches the event to one or more workflow definitions
- WM submits workflow request to the Workflow Engine
- Workflow engine starts a workflow instance on one of many Processing Node
- Workflow instance execution metadata is saved to repository
- WM client monitors execution, retrieves final metadata



CAS Workflow Manager

Features

- Separate standard interfaces for workflow engine and workflow repository
- Out-of-the-box implementations, may be overridden by custom components
- XML-based workflow definition language
- Optional definition of pre/post conditions for workflow execution
- Shared metadata context for all tasks within a workflow instance
- Scalability:
  - Client can submit workflow request to any number of workflow managers
  - Workflow manager can instantiate workflow instances on any number of processing nodes, on local or remote servers

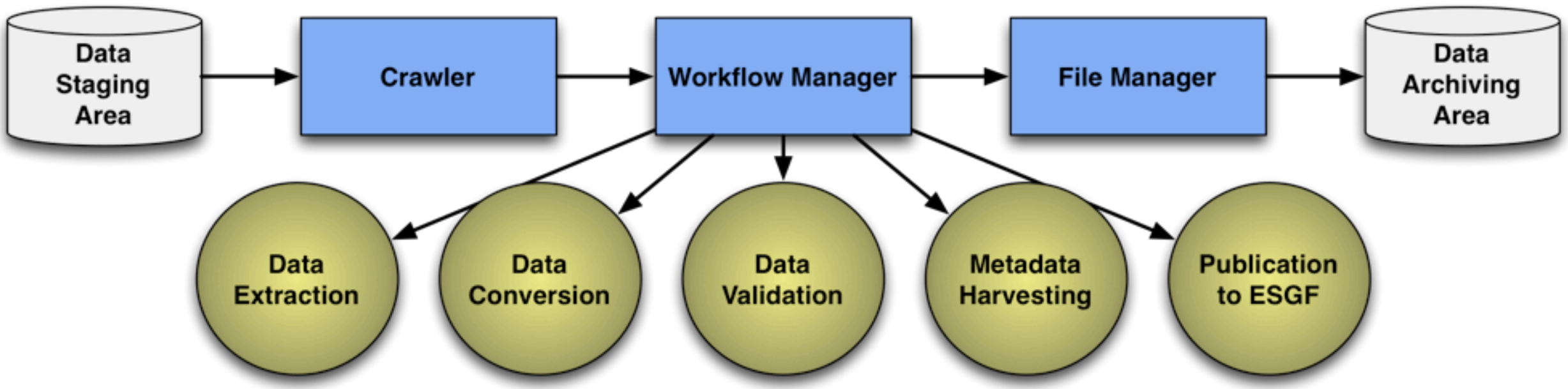
NASA lingo:

- "Process Control System" or PCS: a data processing pipeline expressed as workflow
- "Program Generation Executable" or PGE: a single task within a PCS

CMAC draft Obs4MIPs pipeline

CMAC is working on an automatic, deployable data processing pipeline for Obs4MIPs:

- OODT Crawler setup to automatically detect new data products:
  - Level-2 HDF files, images, or NcML descriptors of data stored on OpenDAP servers
- OODT Crawler sends "publish" event to Workflow Manager
- OODT Workflow Manager executes workflow instance composed of following tasks:
  - Data extraction through mission specific algorithm that conforms to general API
  - Conversion of data to CMIP format (NetCDF/CF)
  - Validation via CMOR-light checker
  - Metadata harvesting
  - Publication to ESGF
- OODT File Manager archives products into final CMIP standard directory structure



Summary & Conclusions

- Science projects are increasingly facing “Big Data” challenges similar to e-commerce:
  - Climate science: XB expected from next generation sensors and models
  - Radio astronomy: next generation telescopes to collect hundreds of TB per sec

- These challenges cannot be met by building timely, expensive in-house solutions, rather, science projects should leverage proven, open source frameworks that can be customized and extended to the project specific needs

- Apache SF includes many freely available technologies that can be used in eScience:
  - Solr provides general, powerful, scalable search capabilities
  - OODT is a framework of software components that can be combined to build automatic, flexible data processing pipelines including data movement, metadata extraction, data archiving & product delivery

- The Earth System Grid Federation is currently using Solr and OODT to serve the climate change community world-wide, and working on even more extensive usage in the future to automatize the distribution of large amounts of NASA observations.

- In the future, ESGF is looking at leveraging Solr and OODT to expand to other science domains (biology, radio-astronomy, health sciences, ...)

Questions ?