Design Of A Webservice and Application For Quick Easy Access To Subsets Of Petabytes Of Air Quality Data Todd Plessel¹, Matt Freeman¹, and Jim Szykman²

¹Lockheed Martin, Environmental Modeling and Visualization Laboratory, National Computing Center, Research Triangle Park (RTP), NC 27711, USA ²Environmental Sciences Division, National Exposure Research Laboratory, Office of Research

CALIPSOSubset.c

double dataMinimur

ouble gcMaximum

unsigned int mask

int propagateDown;

ilterParameters;

CALIPSO L2 05KMAPRO,

CALIPSO L2 05KMAPRO,

nst int found =

const int isCAD =

E (isCAD)

AND2(entry->qcVariable,

if ($\operatorname{gcMinimum} < 0.0$)

onst char* gcVariable

CALTPSO L1. CALTPSO L2 05KMAPRO, et

Minimum valid value for this variable

Name of data variable to filter.

issing value for this variable.

'* Name of data variable to filter by.

0 then use variable levels.

* accept if 0, else apply gcMinimum,gc

/* Propagate the bad/worst qc value from th

* above layers downward before filtering

nimum OC value to accept.

imum OC value to accept

Variables named in this table get filtered by their named QC variables

inction <u>QC</u> Flag 532", 0, 0.0, 0.0, 0xffffffec, 0 /* Clear 0,1,4

arch table for variable name and apply each filter variable foun (index = 0; AND2(ok, filterTable[index].variableName); ++index

ccording to recommendations of the NASA Langley CALIPSO Team 2012-0

tic const FilterParameters filterTable[] = {

AD Score", 0, -100.0, -20.0, 0, 0

ND2(swath->type == entry->type

double qcMinimum = entry->qcMinimum; **double** gcMaximum = entry->gcMaximum

gcMaximum = -swath->minimumCADScore

qcMinimum = swath->minimumCADScore;

'Extinction_Coefficient_532", -0.2, 2.5, MISSING_VALUE,

xtinction Coefficient 532", -0.2, 2.5, MISSING VALUE,

onst FilterParameters* const entry = filterTable + index

strcmp(variable, entry->variableName) == 0);

const int qcLevels = entry->qcLevels ? entry->qcLevels : levels;

! strcmp(entry->qcVariable, "CAD Score"));

bk = filterDataByQC(swath->fileId, data, points, levels,

entry->missingValue

entry->propagateDown

nst char* const uncertainty = uncertaintyVariable(variable);

for (index = 0; ! found && index < swath->variableCount; ++index

-1000.0, 1000.0, MISSING VALUE,

swath->maximumUncertainty, 0, 0);

uncertainty, levels, 0.0,

found = strcmp(swath->variables[index], uncertainty) == 0

ok = filterDataByQC(swath->fileId, data, points, levels,

readProfileElevations(swath, variable, points, levels);

L2 profile data within 180m of surface (first 3 layers)

CALIPSO L2 05KMAPRO, CALIPSO L2 05KMCPRO

CALIPSO L2 05KMALAY, CALIPSO L2 05KMCLAY, CALIPSO_L2_01KMCLAY, CALIPSO_L2_333MCLAY,

erDataNearSurface(data, elevations, points, levels)

and around the world

CMAQ grid paran

ubsetProfileDataAndElevations(swath, data, elevations,

2012-02-09 Per NASA Langley CALIPSO Team:

s possibly invalid so filter it out!

CALIPSO L2 VFM)))

IN8(swath->type,

East (CDC)

Weighted (1/r2)

entry->qcVariable,

Filter points with associated absolute uncertainty

swath->maximumUncertainty. E.g., 99.

double* const elevations

result = swath->data;

= elevations != (

* Adjust QC range by swath->minimumCADScore w/ appropriate sign:

entry->dataMinimum, entry->dataMaximum,

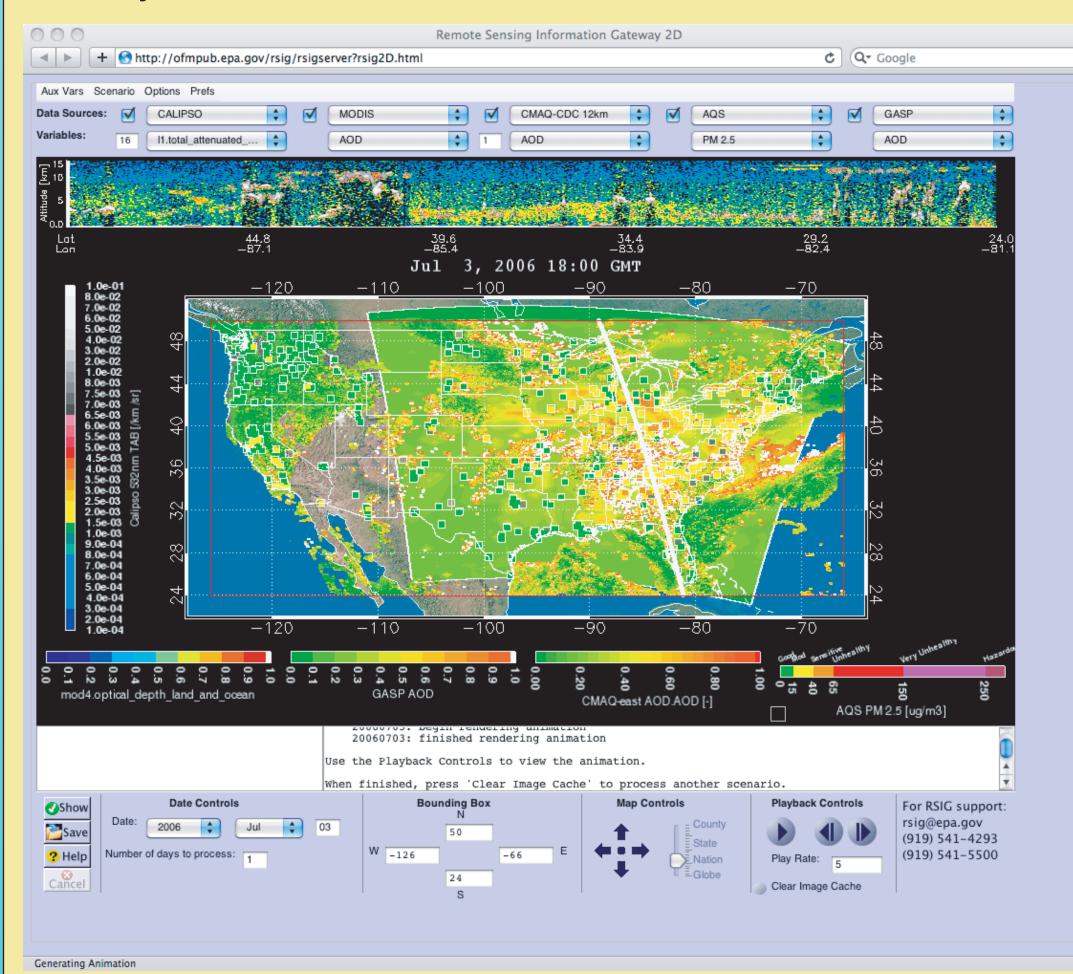
qcLevels, qcMinimum, qcMaximum, entry->mask

ABSTRACT:

EPA's Remote Sensing Information Gateway (RSIG, http://www.epa.gov/rsig) is a widely used free applet and web service for quickly and easily retrieving, visualizing, and saving user-specified subsets of atmospheric data - by variable, geographic domain, and time range.

Petabytes of available data include thousands of variables from a set of NASA and NOAA satellites, aircraft, ground stations, and EPA air-quality models.

We describe the architecture and technical implementation details of this successful system with an emphasis on achieving convenience, high-performance, data integrity and security.

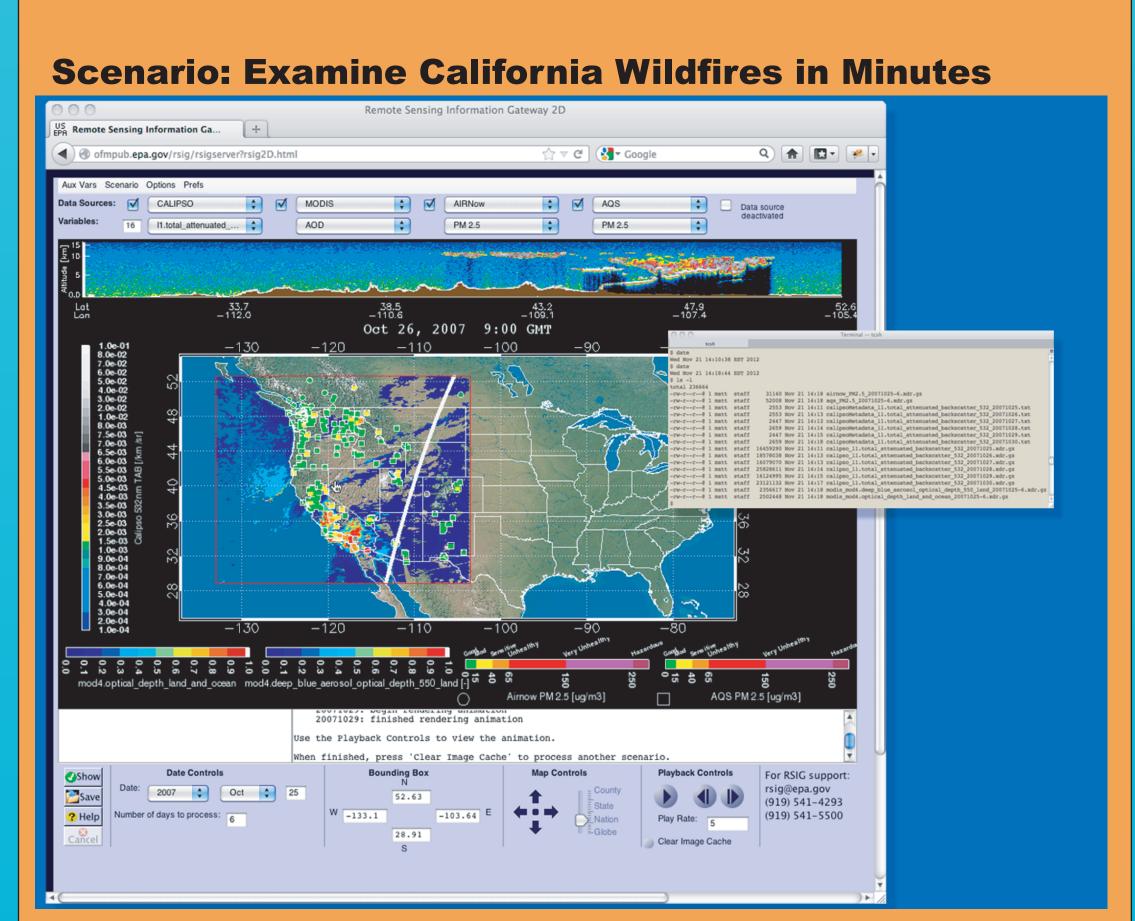


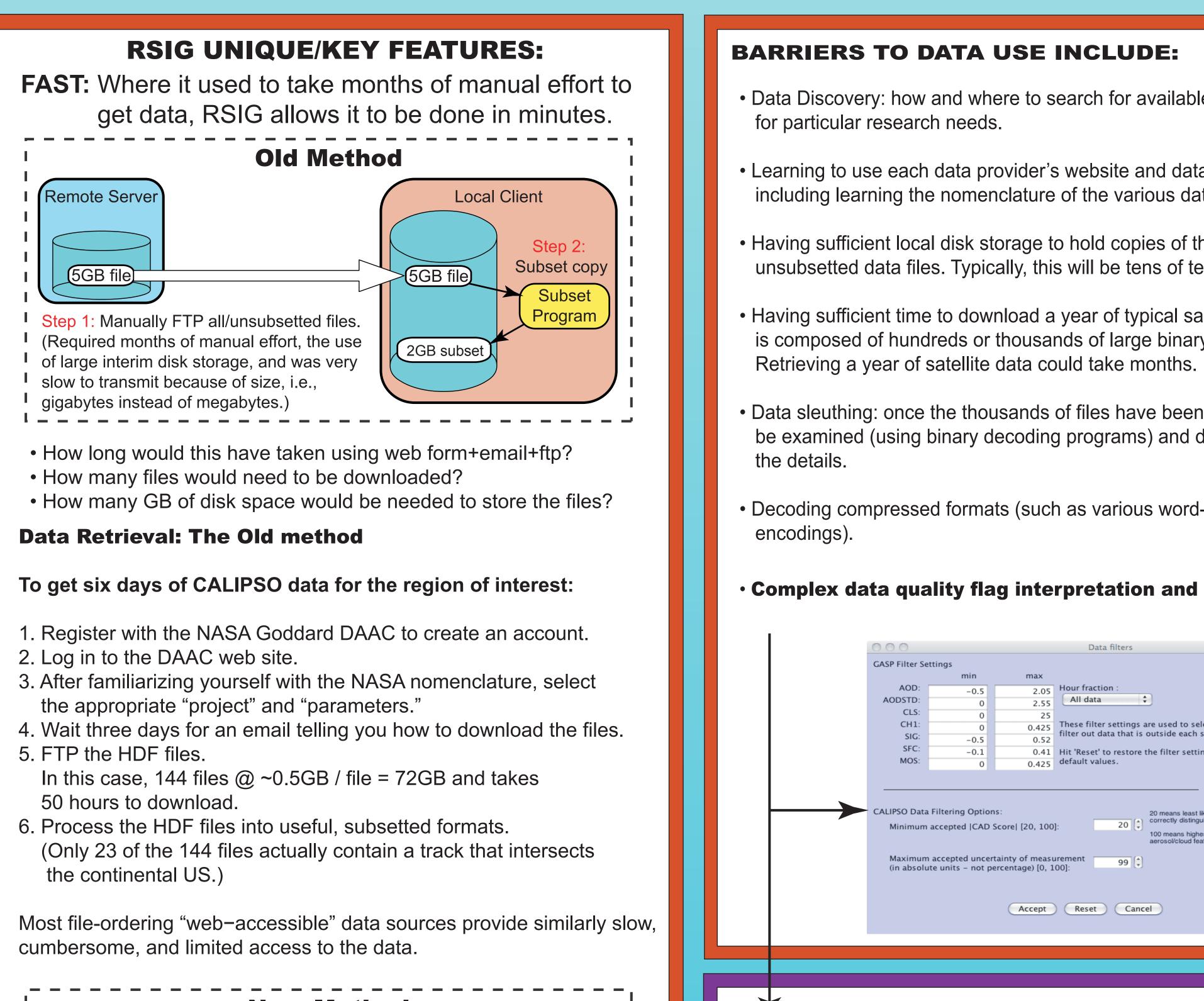
BACKGROUND/MOTIVATION:

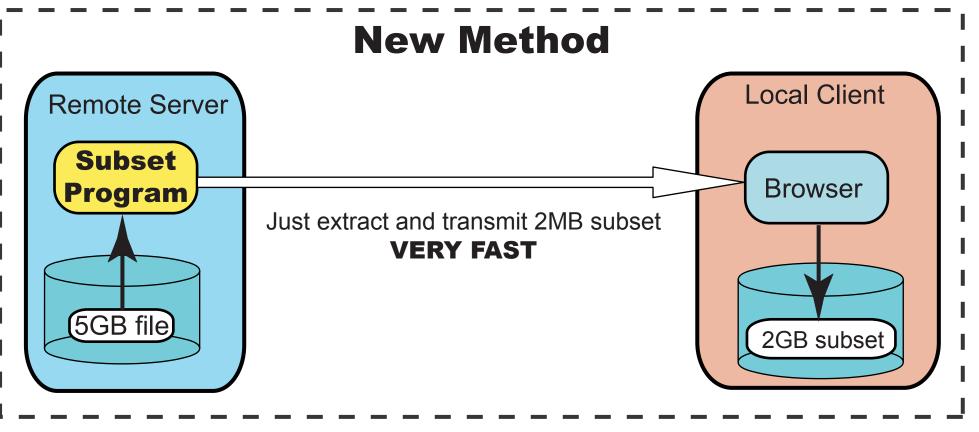
Atmospheric researchers require access to measured data from ground stations, aircraft, and satellites for model evaluation and analysis including exceptional events such as large-scale wildfires.



Petabytes of Air Quality (AQ) data are stored at centers around the country, such as NASA's Distributed Active Archive Centers (DAAC) and NOAA's National Climatic Data Center. However, access to these data is often prohibitively difficult, time-intensive, and requires significant staff and computer resources by the consumer.







Data Retrieval and Vis: The RSIG Way – Quick & Easy

Thu Feb 7 15:52:48 EST 2008 Thu Feb 7 15:57:56 EST 2008 \$ ls -asl 64 -rw-rw-r-1 plessel visstaff 31978 Feb 7 15:55 airnow_PM2.5_20071023-6.xdr.gz 0 -rw-rw-r-- 1 plessel visstaff 0 Feb 7 15:55 aqs_PM2.5_20071023-6.xdr.gz 473272 -rw-rw-r-1 plessel visstaff 242314947 Feb 7 15:55 calipso_TAB532nm_20071023-6.xdr.gz 16 -rw-rw-r-1 plessel visstaff 5319 Feb 7 15:55 goes-bb_PM2.5_20071023-6.xdr.gz 69000 -rw-rw-r-1 plessel visstaff 35298901 Feb 7 15:58 modis2_COT_20071023-6.xdr.gz 9904 -rw-rw-r-1 plessel visstaff 5070301 Feb 7 15:55 modis_AOD_20071023-6.xdr.gz

Data Retrieval and Subsetting: The RSIG Method \$ uncompress *.gz

s ls -asl

Uncompressed XDR files total just 1GB.

\$ head -12 airnow_PM2.5_20071023-6.xdr AIRNOW 1.0 2007-10-23T00:00:00-0000 # data dimensions: timesteps stations 144 194 # Variable names # Variable units: # MSB 32-bit integers ids[stations] and # IEEE-754 32-bit reals sites[stations][2=<longitude,latitude>] and # IEEE-754 32-bit reals data[timesteps][stations]:



Visualization

Just five minutes to stream and visualize six days worth of data! + CALIPSO LIDAR Backscatter

- + MODIS Aerosol Optical Depth
- + MODIS Cloud Optical Thickness
- + Airnow PM 2.5
- + NESDIS Biomass Burning PM 2.5

Data Download

Saving the data subset to local disk:

Just five minutes to stream & save full-resolution subsetted data and yielded under 250MB in compressed simple-format files.

224 –rw–rw–r– 1 plessel visstaff 114367 Feb 7 15:55 airnow_PM2.5_20071023–6.xdr 1615560 -rw-rw-r-- 1 plessel visstaff 827165389 Feb 7 15:55 calipso_TAB532nm_20071023-6.xdr 128 -rw-rw-r-1 plessel visstaff 61444 Feb 7 15:55 goes-bb_PM2.5_20071023-6.xdr 292400 -rw-rw-r-1 plessel visstaff 149704919 Feb 7 15:58 modis2_COT_20071023-6.xdr 36232 -rw-rw-r-1 plessel visstaff 18548851 Feb 7 15:55 modis_AOD_20071023-6.xdr

ASCII headers and XDR binary arrays are efficient and easy to read/parse:

For more information on RSIG and available data, visit our website: http://www.epa.gov/rsig/

Accept Reset Cancel

- Data Discovery: how and where to search for available data useful
- Learning to use each data provider's website and data-ordering tools, including learning the nomenclature of the various data products.
- Having sufficient local disk storage to hold copies of the ordered, unsubsetted data files. Typically, this will be tens of terabytes.
- Having sufficient time to download a year of typical satellite data, which is composed of hundreds or thousands of large binary data files.
- Data sleuthing: once the thousands of files have been acquired they must be examined (using binary decoding programs) and deciphered to learn
- Decoding compressed formats (such as various word-size integer

Complex data quality flag interpretation and filtering

- All data 🛟 ese filter settings are used to selectively ter out data that is outside each specified range. 0.41 Hit 'Reset' to restore the filter settings to the 0 means least likely (but better than a 50/50 chance of correctly distinguished aerosol/cloud feature). Minimum accepted |CAD Score| [20, 100]: 100 means highest likelyhood of correctly distinguished aerosol/cloud feature.
 - Accept Reset Cancel
 - Webservice Security: Safe and Secure Parsing The PERL-CGI webserver scripts are designed with all known security best practices, safe and correct parsing. • The code has passed all reviews by EPA and NASA security experts. • The system has passed all security tests at EPA and NASA. Examples of some security techniques are: Untainting all query string input data • Restricting the environment variables Non-shell spawning processing as shown in the code excerpts below
 - rsigserver (PERL CGI) #!/usr/bin/perl -wT
 # Enable PERL's warning (-w) and tainted input data-flow tracing (-T). package main; \$| = 1; # Turn off output buffering so messages appear in correct order. alarm(3600); # Kill process after 1 hour. # Restrict PATH and ENV. delete @ENV{ 'PATH', 'IFS', 'CDPATH', 'ENV', 'BASH_ENV' } use CGI qw/ untaint /; # Load Safe untainting CGI routines. \$CGI::DISABLE_UPLOADS = 1; # Disable uploads. \$CGI::POST_MAX = 1024; # Limit posts to 1024 bytes. \$CGI::DISABLE_OFLOADS = 1; # DISABLE uploads. \$CGI::DOST_MAX = 1024; # Limit posts to 1024 bytes. my §query = new CGI; # Parse QUERY_STRING. %ENV = (); # Unset all environment variable my @names = \$query->param; # Extract names of parameters. *#* Unset all environment variables
 - cor (my \$parameter = 0; \$parameter < \$count; ++\$parameter)</pre> \$_ = \$names[\$parameter]; s/[^A-Za-z]/ /go; squery->param(\$option); s/[^\w\-.,:\/]/_/go; \$_ is tainted. my \$value = \$_;
 my \$lowercase_option = lc(\$option);
 my \$lowercase_value = lc(\$value); # \$value is untainted. if (\$parsers{ \$lowercase_option }) {
 \$result = \$parsers{ \$lowercase_option }->(\$lowercase_value); print STDERR "\nInvalid guery string.\n";
 - sub execute_command {
 my \$command = shift;
 my \$command = command = comman if (\$command =~ $m \#^{(/ | w-|+/ | w -/., ':?=&|+)}$) \$command = \$1; %ENV = (); # Unset all environment variables prior to popen. ny \$pid = open(the_pipe, "-|"); if (! defined(\$pid)) {
 die "Couldn't open pipe to subprocess";
 } elsif (\$pid) { # Parent process.
 - while (<the_pipe>) {
 print; close(the_pipe) or die \$!
 \$result = ! \$?;
 - } else { # Child process.
 exec(\$command) or die "can't exec program: \$!"; else {
 print STDERR "\n\$0: command contains invalid characters.\n"; return \$result;
- SIG was created to remove these barriers to use and provide additional val added such as regridded data to modules used by atmospheric researchers at EPA
 - 2
 9.81
 9.81

 33
 deg
 R:
 287.04
 J/kg/K

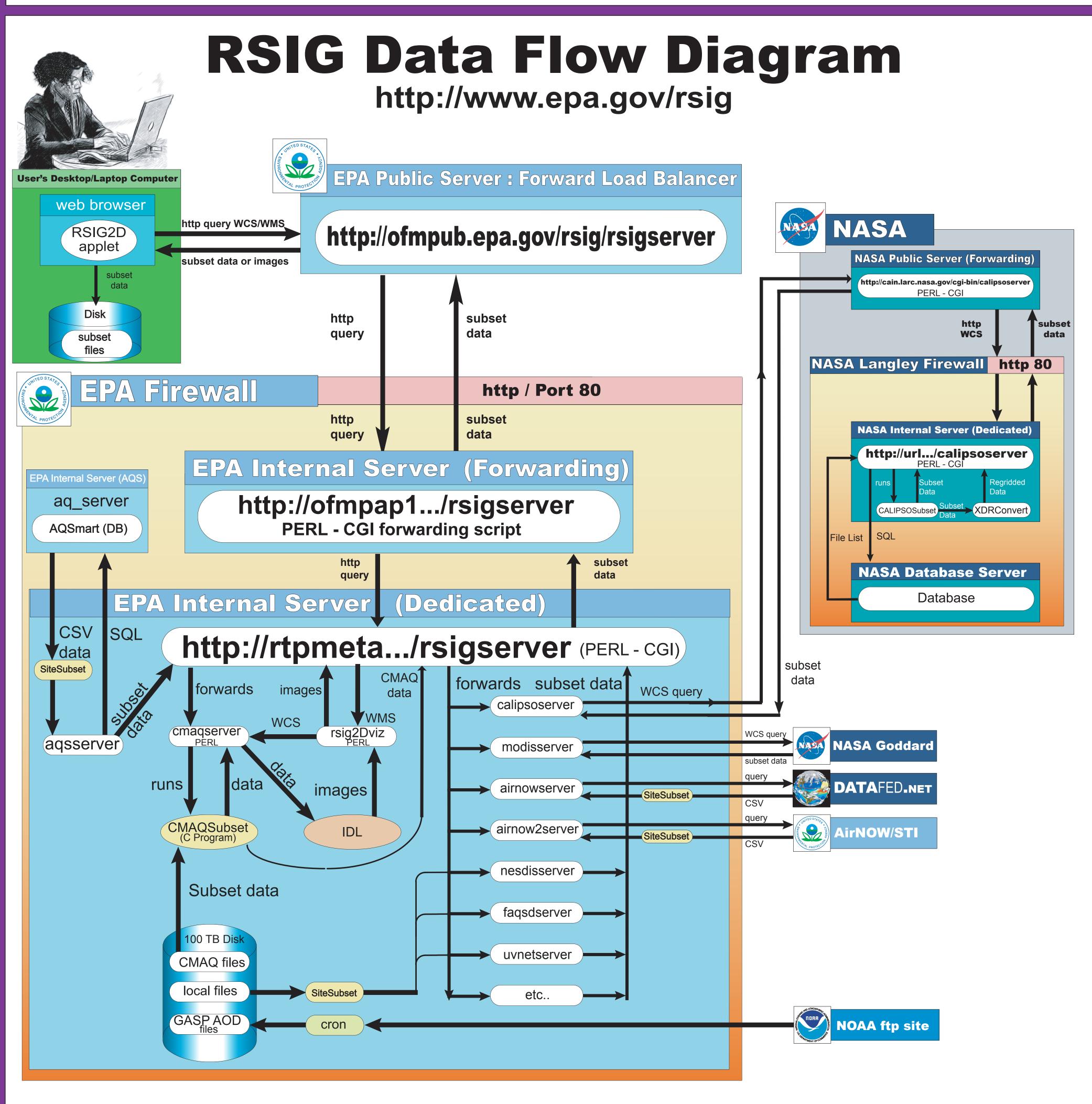
 45
 deg
 A:
 50
 K

 -97
 deg
 T0:
 290
 K

 -97
 deg
 P0:
 100,000
 Pa
 P_BET : P_GAM : XCENT :

ARCHITECTURE:

- Public Components
- and interoperability with external software applications ("mash-ups").



IMPLEMENTATION COMPONENTS:

Behind the scenes of the public components, there are a chain of webservices invoking data subsetting programs that read the data files needed for a request.

CONCLUSION:

- over the internet.
- capabilities added every year.

ACKNOWLEDGMENTS:

We gratefully acknowledge the contributions of many who have worked with the RSIG team to provide access to data sets. In particular we would like to acknowledge the support of Danny Mangosing (SSAI), Mark Vaughan, Jason Tackett, Pamela Rinsland, and Chip Trepte at NASA LaRC, and Ed Matsuoka, Steve Kempler, Cid Praderas (Sigma Space), and Greg Ederer (Sigma Space) at NASA GSFC, and Shobha Kondragunta, Pubu Ciren, and Chuanyu Xu at NOAA/NESDIS/STAR.

ABSTRACT REFERENCE NUMBER: 1482803 PAPER REFERENCE NUMBER: IN31A-1498

• Applet: Researchers primarily use the RSIG2D applet whose simple graphical user interface and visualization features enable quick and easy selection, retrieval, visualization, and saving of subsets of air-quality-related data from a variety of sources, including NASA, NOAA, and EPA. • Webservice: The applet uses the free, publicly accessible webservice - rsigserver - to retrieve specified data variables subsetted by longitude-latitude rectangle and date-time range. The rsigserver webservice is based on OGC-WCS for compatibility, data discovery (REQUEST=GetCapabilities)

• For large daily-generated datasets, these data-specific webservice applications and subsetters are installed at the data provider site. • For small or static datasets, the data and processing is stored on a dedicated server inside the firewall at EPA. • The webserver applications are PERL-CGI scripts that safely parse the query string, issue SQL to a database for the list of data files needed, and invoke the subsetter programs to efficiently read the files - extracting and streaming the subset of data requested back to the EPA server - to be rendered into images for display in the user's web browser or else the data is streamed back to the user's computer and saved to their local disk. • Subsetter programs are designed for correct data processing, including complex data quality filtering, high performance, and efficiency.

• Development is driven by EPA research needs, as determined by the project's principal investigator and his colleagues. • Intuitive graphical interface allows users to quickly and easily access and compare selected datasets from massive, remote data repositories. • Demonstrates the power of collaborative development across Federal Agencies, e.g., NASA and EPA have worked closely together to develop and deploy efficient and secure data server and subsetting codes at the data sources, reducing by orders of magnitude the volume of data streamed

• Since project inception in 2005, RSIG has been used by over 100 institutions world-wide and its development continues to evolve with new data and

United States Environmental Protection Agency

