End to End Services Under the Hood

Cast of Thousands
E2E Services Goals

1. Availability of core services for all appropriate data
2. Consistent, high-quality User Experience
3. Scaling out without code: database-driven
The Cast - Software

● Earthdata Search: the Front Door
● Common Metadata Repository (CMR): the Nexus
  ○ UMM-Collection
  ○ UMM-Granule
  ○ UMM-Service
  ○ And introducing...UMM-Variable
  ○ CMR Auto-population
● Metadata Management Tool: the Fixer
● Service Broker: the Go-Between
● The Services: where the magic happens
  ○ OPeNDAP
  ○ HEG
  ○ GDAL
  ○ ...

The Cast - People

● Users
  ○ Science researchers looking for Analysis Ready Data
  ○ Applications developers: ArcGIS, Panoply, nco, ...

● DAAC Data Curators

● EED2 Teams
  ○ CMR
  ○ MMT
  ○ SDPS Services / CSI
  ○ (OPeNDAP - if needed)
  ○ (HDF - if needed)

● System Architects

● (Join us…)
The Cast - Data

- Gridded Data
  - Served via OPeNDAP
  - Served via other REST-ish Web Services
    - THREDDS Data Service
    - Web Coverage Service
    - others...

- Swath data
  - Served via OPeNDAP (variable subsetting, reformatting)
  - Served via other Web Services
End to End Services

Start

1. Data Provider
2. Archive
3. CMR
4. WCS
5. Earthdata Search
6. GIS
7. WCS-OPeNDAP Broker
8. Transformed data

End
Adding useful services begins with data products that are easy to work with:

- Compliant with standards and conventions
  - NetCDF, CF, ESO RFCs
- Populated with useful metadata
  - Variable long_names
  - Variable units

A working group is developing a Data Product Developers’ Guide to help algorithm developers design and implement data products that are easy to work with.
Data Product Developers’ Guide

Introduction
  Intended Audience
  Purpose
  Scope

Data Product Design Process
  Requirements: determining what the user community needs
  Design: What constitutes a data product design?
  Implementation: Creating sample data products
  Testing: Evaluating sample products against requirements
  Review: Independent evaluation of the data product (Downs)

Selecting a Data Product Name (Dana Ostrenga)

Data Product Versions (Elliot Sherman)

Selecting a Data Format (John Moses)
  ESDIS Standards Office
  Recommended formats: netCDF4/CF, HDF5, HDF-EOS5
  Other ESO-approved formats: ICARTT, ...
  Deprecated formats: HDF-4, HDF-EOS2, netcdf3
  Other Formats
CMR UMM-Var Auto Population

Function (CMR): Storage of collection, granule, variable and service metadata

UI (MMT): Review/edit/update of metadata records, either single record or bulk update

- Manual population of UMM-C, UMM-S records via MMT forms (or via the API)
- Auto-pop function of UMM-Var - single record update
- Bulk update function of UMM-Var, C - multiple record update
3 Metadata Management Tool

- Population of UMM-Service
- Refinement of UMM-Variable
  - Single Record
  - Bulk Update
  - Manage auto-population
Earthdata Search: Presenting Customization Options

- Variable Selection
- Spatial Constraints
- Temporal Constraints
- Format
- Other transformations (e.g. reprojection)
Earthdata Search: Variable Selection

<table>
<thead>
<tr>
<th>Variable Selection</th>
<th>Customizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surface Pressure</td>
<td>Aqua AIRS Level 3 Daily Standard Physical Retrieval (AIRS+AMSU) V006 (AIRX3STD)</td>
</tr>
<tr>
<td>Carbon Monoxide</td>
<td>UMM-Var</td>
</tr>
<tr>
<td>Cloud Frequency</td>
<td>science</td>
</tr>
<tr>
<td>Cloud Top Pressure</td>
<td>keywords</td>
</tr>
<tr>
<td>Cloud Top Temperature</td>
<td>long_name</td>
</tr>
<tr>
<td>Cloud Vertical Distribution</td>
<td></td>
</tr>
<tr>
<td>Geopotential Height</td>
<td></td>
</tr>
<tr>
<td>Humidity</td>
<td></td>
</tr>
<tr>
<td>Surface Pressure</td>
<td></td>
</tr>
</tbody>
</table>
Earthdata Search: Spatial Subset Constraints

- Spatial ROI from search carried over to Spatial Subset
- Further spatial refinement allowed for datasets with global granules
Earthdata Search: Submitting the customization request

- **Granule-level OPeNDAP URLs**
  - OnlineResourceURL (not OnlineAccessURL)
  - Service endpoint at granule level (without any response suffix)

- **Data Collection OPeNDAP URLs**
  - Pattern substitution to convert granule OnlineAccessURLs into Granule-level OPeNDAP URLs
“DEMO”
WCS GetCoverage (planned work)

- Current OUS broker
  - Future work will enable Earthdata Search to issue requests using WCS GetCoverage directly to the WCS-OPeNDAP broker.
WCS-OPeNDAP Broker

- Converts spatial/temporal constraints into OPeNDAP array indices for a data product
  - Uses coordinates info from UMM-Variable
- Produces valid OPeNDAP subset requests for selected variables in a set of granules
Usable End Products...

Output of E2E subsetting should be **Analysis-Ready Data**:

- Compliant with standards and conventions
  - CF
  - ESO RFCs
  - Data Product Developer’s Guide
- Usable within target analysis tools
  - ArcGIS
  - Panoply
  - nco
  - ...
<table>
<thead>
<tr>
<th>Name</th>
<th>Long Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>icesat.nc</td>
<td>icesat.nc</td>
<td>Local File</td>
</tr>
<tr>
<td>gt2l</td>
<td>gt2l</td>
<td>—</td>
</tr>
<tr>
<td>lan...</td>
<td>gt2l/land_ice_height</td>
<td>—</td>
</tr>
<tr>
<td>...</td>
<td>Elapsed GPS sec</td>
<td>1D</td>
</tr>
<tr>
<td>l...</td>
<td>Latitude</td>
<td>1D</td>
</tr>
<tr>
<td>l...</td>
<td>Longitude</td>
<td>1D</td>
</tr>
<tr>
<td>...</td>
<td>Vertical Geolocation</td>
<td>1D</td>
</tr>
</tbody>
</table>

:-(
Usable End Products (?)

Variable "sigma_geo_h"
In file "icesat.nc"
Var full name: gt2l/land_ice_height/sigma_geo_h

```
float sigma_geo_h(delta_time=3183);
    _FillValue = 3.4028235E38f; // float
    contentType = "qualityInformation";
    long_name = "Vertical Geolocation Error";
    description = "Total vertical geolocation error"
    source = "section 3.10"
    coordinates = "delta_time, latitude, longitude"
    units = "meters"
```
Usable End Products (?)

Variable "sigma_geo_h"

In file "icesat.nc"

Var full name: gt2l/land_ice_height/sigma_geo_h

float sigma_geo_h(delta_time=3183);
    _FillValue = 3.4028235E38f; // float
    :contentType = "qualityInformation";
    :long_name = "Vertical Geolocation Error";
    :description = "Total vertical geolocation error";
    :source = "section 3.10";
    :coordinates = "delta_time, latitude, longitude";
    :units = "meters";
Usable End Products (yay!)

<table>
<thead>
<tr>
<th>Name</th>
<th>Long Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>icesat.nc</td>
<td>icesat.nc</td>
<td>Local File</td>
</tr>
<tr>
<td>gt2l</td>
<td>gt2l</td>
<td>—</td>
</tr>
<tr>
<td>lan...</td>
<td>gt2l/land_ice_height</td>
<td>—</td>
</tr>
<tr>
<td>...</td>
<td>Elapsed GPS sec...</td>
<td>1D</td>
</tr>
<tr>
<td>l...</td>
<td>Latitude</td>
<td>1D</td>
</tr>
<tr>
<td>l...</td>
<td>Longitude</td>
<td>1D</td>
</tr>
<tr>
<td>...</td>
<td>Vertical Geolocat...</td>
<td>1D</td>
</tr>
<tr>
<td>test1.nc</td>
<td>test1.nc</td>
<td>Local File</td>
</tr>
<tr>
<td>gt2l</td>
<td>gt2l</td>
<td>—</td>
</tr>
<tr>
<td>lan...</td>
<td>gt2l/land_ice_height</td>
<td>—</td>
</tr>
<tr>
<td>...</td>
<td>Elapsed GPS sec...</td>
<td>1D</td>
</tr>
<tr>
<td>l...</td>
<td>Latitude</td>
<td>1D</td>
</tr>
<tr>
<td>l...</td>
<td>Longitude</td>
<td>1D</td>
</tr>
</tbody>
</table>

Variable "sigma_geo_h"
In file "test1.nc"

Var full name: gt2l/land_ice_height/sigma_geo_h

float sigma_geo_h(delta_time=3183);
  _FillValue = 3.4028235E38f; // float
  :contentType = "qualityInformation";
  :long_name = "Vertical Geolocation Error";
  :description = "Total vertical geolocation error in meters";
  :source = "section 3.10";
  :coordinates = "latitude longitude";
  :units = "meters";
Usable End Products (yay!)
Data Stewardship Questions to Ponder:

1. When is it OK to improve on the data provider’s data product?
2. What kinds of improvements are legit?
What do the DAACs need to do to enable services for their collections?

1. Determine which data and services need to be accessible in CMR
2. For each service, determine how the data are organized relative to the service end-point
3. Describe the service options for those data: i.e., output formats, projections
4. Assemble information about the service: name, long name, type, version, any related info (user guides, web portals, etc.), access constraints, quality info, contact info
5. Complete the Service record form and make CMR associations between the service record and the collections
6. See Metadata Quality Team for assistance and review
How to populate a UMM-S record

1. Enter Service Information into the first form (e.g. Name, Long Name, Type, Version, Description
2. Enter Related URLs (currently OPeNDAP URL, User Guides, web portals)
3. Enter Service Keywords (choose form pull-down lists)
4. Enter Service Organization Info: (Service Provider Name, Contact Groups/Persons/Contact Type (e.g. telephone, email, etc., hours of service))
5. Enter Service Options (e.g. spatial, temporal, variable subsetting, formats, projections, etc.)
Example UMM-S record (From 18.2)

```
"Name":"AIRX3XXX.006",
"LongName":"AIRS/Aqua L3 Daily Standard Physical Retrieval (AIRS+AMSU) 1 degree x 1 degree V006.",
"Description":"Access the AIRS/Aqua Level 3 daily standard physical retrieval product (Without HSB) through OPeNDAP",
"Type":"OPeNDAP",
"Version":"1.9",
"AccessConstraints":"None",
"UseConstraints":"None",
"RelatedURLs":
[
  {
    "Description":"OPeNDAP Service for AIRX3XXX.006",
    "URLContentType":"DistributionURL",
    "Type":"GET SERVICE",
    "Subtype":"OPeNDAP DATA",
    "URL":"https://acdsc.gesdisc.eosdis.nasa.gov/opendap/Aqua_AIRS_Level3/"
  },
  {
    "Description":"User Guide",
    "URLContentType":"PublicationURL",
    "Type":"VIEW RELATED INFORMATION",
    "Subtype":"USER'S GUIDE",
    "URL":"http://docserver.gesdisc.eosdis.nasa.gov/repository/Mission/AIRS/3.3_ScienceDataProductDocumentation/3.3.4_ProductGeneration/"
  }
],
"ServiceKeywords":
["DATA MANAGEMENT/DATA HANDLING","TRANSFORMATION/REPROJECTION"],
"ScienceKeywords":
["EARTH SCIENCE","ATMOSPHERE","ATMOSPHERIC CHEMISTRY","CARBON and HYDROCARBON COMPOUNDS","Methane"],
"Variables":null
```
How to populate a UMM-Var record

**Manual steps (18.3)**

1. Determine variables to be populated from a collection
2. Group variables into an ordered set (use the granule metadata as a guide) or open a granule from the collection in Panoply
3. Enter Variable Information into the form (e.g. Name, Long Name, Definitions, Data Type, Dimensions, Sets, Scale and Offset, Variable Type)
4. Add associations with the collection to CMR

**Auto-pop - planned to launch via MMT (18.4)**

**Bulk update - planned to launch via MMT (18.4)**
Example UMM-Var record (from 18.2)

```
"Dimensions":
  [{
    "Name":"StdPressureLevel",
    "Size":24
  },
  {
    "Name":"Latitude","Size":180
  },
  {
    "Name":"Longitude","Size":360
  }],
"Scale":1.0,
"Offset":0.0,
"Sets":
  [{
    "Name":"Data_Fields",
    "Type":"Ascending",
    "Size":146,"Index":15
  }],
"Units":"hpa",
"FillValues":
  [{
    "Value":-9999.0,
    "Type":"ANCILLARY_FILLVALUE"
  }],
"ScienceKeywords":
  [{
    "Category":"EARTH SCIENCE",
    "Topic":"LAND_SURFACE",
    "Term":"SURFACE THERMAL PROPERTIES",
    "VariableLevel":"SKIN TEMPERATURE"
  }],
"Measurements":
"VariableType":"SCIENCE_VARIABLE"
```
To Do

1. Operationalize
2. Add Hyrax-served grid datasets
3. Add ways to auto-pop: thesaurus, text mining, ...
4. Web-Coverage-Service-ize
5. Add more services: TDS, WCS, ...
6. Cumulus-ize
Cloud: In a world...

- Earthdata Search
- netCDF
- WCS
- <Your tool here>

<Your Service Here>

OPeNDAP
Join Us...

(You don't really need to know python)