LaTiS: Standards-based Web Service Interfaces for Diverse Data

Doug Lindholm & Chris Lindholm

University of Colorado Boulder

Laboratory for Atmospheric and Space Physics

doug.lindholm@lasp.colorado.edu | chris.lindholm@lasp.colorado.edu

LaTiS is an Open Source software library that is designed to read from multiple diverse data sources, perform dataset subsetting and transformations, and write the results in a variety of formats. Data providers can deploy LaTiS as a data server (with no additional coding) exposing standards-based web service interfaces including the Heliophysics Application Programmer's Interface (HAPI). As a result, data users can avoid data wrangling by accessing the data they need via a single API, improving interoperability.

This poster demonstrates how a LaTiS server can be configured to read datasets from a variety of sources and formats and provide a unified web service interface that multiple clients, including your code, can use to integrate data sources into a single application.

External Data Sources

CDAWeb HAPI Server

Read a dataset from a remote HAPI data service, delegating to its HAPI API.

https://data.ngdc.noaa.gov

NetCDF

Combine a set of remote yearly netCDF files into a single virtual dataset.

Code for this server demo

Dataset Descriptors

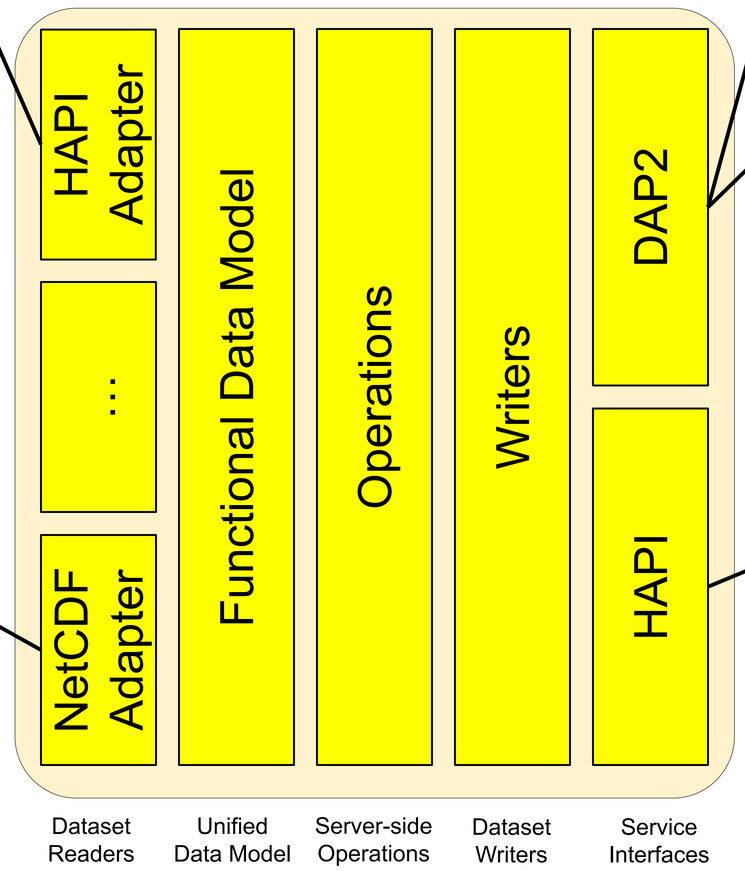
```
<dataset id="ac_k0_swe"
         temporalCoverage="2017-07-01/">

  <source uri="https://cdaweb.gsfc.nasa.gov/hapi/"/>

  <adapter class="latis.input.HapiCsvAdapter"
           id="AC_K0_SWE"/>

  <function>
    <scalar id="time"
            type="string"
            units="yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"
            size="24"
            class="latis.time.Time"/>
    <tuple>
      <scalar id="Np"
              title="Solar Wind Proton Number Density"
              type="double"
              units="#/cc"
              missingValue="999.9"/>
      <scalar id="Vp"
              title="Solar Wind Bulk Speed"
              type="double"
              units="km/s"
              missingValue="999.9"/>
    </tuple>
  </function>

</dataset>
```

ac_k0_swe.fdml

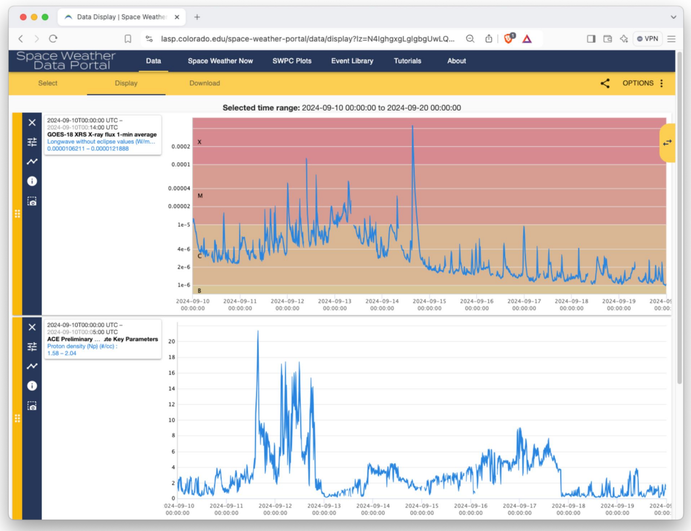
```
<dataset id="noaa_goes18_xrs"
         temporalCoverage="2022-09-02/"
         class="latis.dataset.GranuleAppendDataset">

  <source>
    <dataset>

      <source
uri="https://data.ngdc.noaa.gov/platforms/solar-space-observing-satellites/goes/goes18/l2/data/xrsf-l2-avg1m_science/"/>

      <adapter class="latis.input.GranuleListGenerator"
               pattern="sci_xrsf-l2-avg1m_g18_y%1$tY_v2-2-0.nc"
               start="2022"
               step="P1Y"/>

      <function>
        <scalar id="time"
                type="string"
                units="yyyy-MM-dd'T'HH:mm:ss.SSS'Z'"
                class="latis.time.Time"/>
        <scalar id="uri"
                type="string"/>
      </function>

    </dataset>
  </source>

  <adapter class="latis.input.NetcdfAdapter"/>

  <function>
    <scalar id="time"
            units="seconds since 2000-01-01T12:00:00"
            type="double"
            class="latis.time.Time"/>
    <tuple>
      <scalar id="xrsa_flux"
              title="XRS-A primary average flux"
              type="float"
              units="W/m^2"
              missingValue="-9999"/>
      <scalar id="xrsb_flux"
              title="XRS-B primary average flux"
              type="float"
              units="W/m^2"
              missingValue="-9999"/>
    </tuple>
  </function>

</dataset>
```

noaa_goes18_xrs.fdml

LaTiS



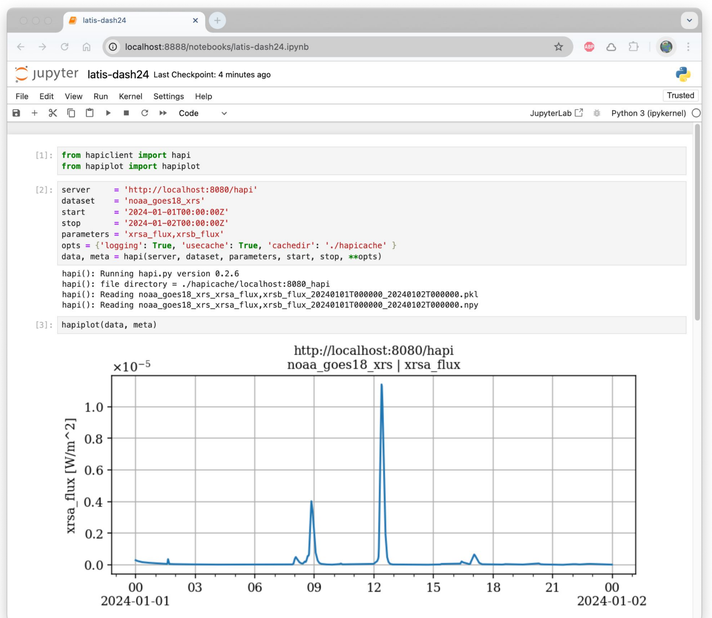
Client applications



Space Weather Data Portal

http://localhost:8080/dap2/noaa_goes18_xrs.csv?time>=2024-01-01&time<2024-01-02&project(time,xrsb_flux)&formatTime("yyyy-MM-dd'T'HH:mm:ss")

Sample LaTiS DAP2 URL



Jupyter Notebook