Supported by NASA Grant NNX17AH27G

Keeping Everyone HAPI: Achieving Interoperability for Heliophysics and Planetary Time Series Data

IN11D-0644

Jon Vandegriff (JHU/APL), Todd King (UCLA), Robert Weigel (GMU), Jeremy Faden (Cottage Systems / University of Iowa), Aaron Roberts (NASA/GSFC), Bernie Harris (NASA/GSFC), Robert Candey (NASA/GSFC), Nand Lal (NASA/GSFC), Scott Boardsen (UMBC), Doug Lindholm (Colorado University / LASP), Thomas Baltzer (Colorado University / LASP), Lawrence Brown (JHU/APL), and Eric Grimes (UCLA)

The ability to universally access timeseries data with one API would provide a foundational element for enhancing science data interoperability in many fields. The Heliophysics Application Programmers Interface (HAPI) offers a simple way to expose time series data online. HAPI standardizes the two key parts of a data service: the request interface and the result format. HAPI is being adopted by multiple data centers within the Planetary and Heliophysics communities (especially among providers of plasma, particle and field data), and it is under consideration as a COSPAR standard.

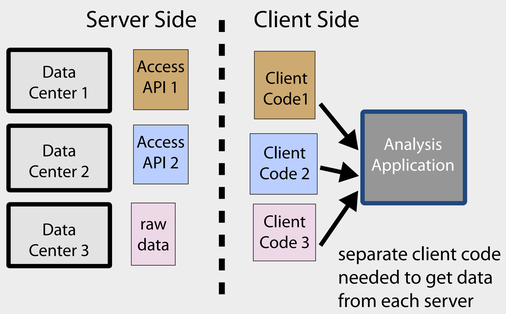
The ease with which various providers have adapted existing servers to create a HAPI-compliant capability shows that it does capture a useful way to represent time series data. Because clients for reading HAPI data are also easy to create, we anticipate significant growth and interest in this emerging standard.

To receive HAPI update emails:
send a request to: `hapi-news@hapi-server.org`

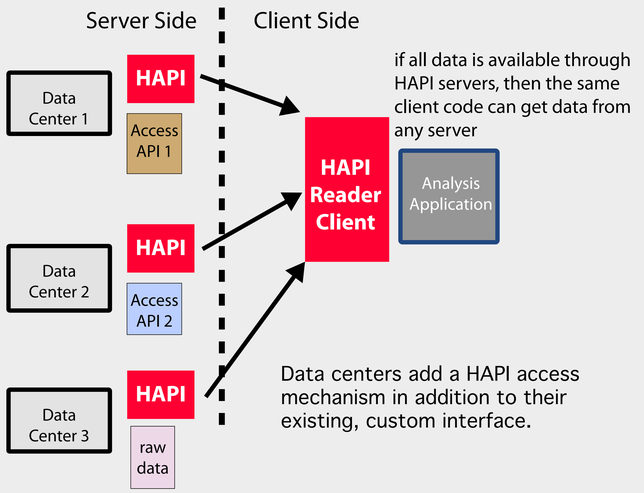
The complete HAPI specification is at GitHub.
`https://github.com/hapi-server/data-specification`

Barrier to Interoperability: many interfaces

For very similar functionality, many data centers have a unique interface.



The Solution: Standardize on the Lowest Common Denominator



Overview of the HAPI Specification

Every HAPI server must implement four endpoints below the root hapi element.

http://example.org/hapi the root URL for the server is a human readable landing page

Each of the four endpoints are directly below the root URL.

capabilities

list the optional elements implemented by the server

arguments: none; return format: JSON

catalog

Try it now with these QR code links to an active server.

list the names of the datasets the server can provide

arguments: none; return format: JSON

info

provides a description of one dataset; the metadata is minimal, but enough to configure a reader

arguments: dataset identifier; return format: JSON

data

(This provides CSV. JSON and binary are also possible.)

streams data content in a format defined by the spec; time values must all be standardized

arguments: dataset identifier, time range, and desired output format; return format: CSV, binary or JSON (details defined by spec)

`json`

`binary`

Key features and details

- The data request interface is very simple (bold shows user inputs of dataset id and time range; multiple lines are shown for clarify):

```
http://datashop.elasticbeanstalk.com/hapi/
    data?id=CASSINI_LEMMS_PHA_CHANNEL_1_SEC&
    time.min=2002-01-02T00:00Z&
    time.max=2002-01-02T03:00Z
```

- Data returned is a stream, hiding the granularity and structure of underlying data.
- Request interface allows asking for a subset of parameters.
- Servers must be able to generate a CSV data stream; JSON & binary output are optional.
- All time values in the data stream are ISO 8601 strings.
- The required metadata is very minimal (just enough to read the data), but it can point to other metadata (SPASE records in Heliophysics, for example).
- Servers can add additional custom metadata keywords with a specific prefix.
- Multi-dimensional data is supported, as are the bin descriptions for spectral data.
- Error codes are defined for many error conditions, including requesting too much data (each server decides how much is too much).

Existing Capabilities

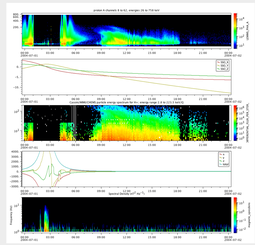
Server Checker - tests the compliance of a server to the HAPI spec.
- very useful for developers!

`http://tsds.org/verify-hapi`

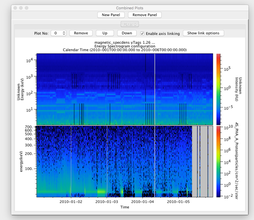
HAPI Clients

Plotting and downloading of HAPI data; note the ability to stack plots of data from different HAPI servers

Autoplot
`http://autoplot.org`

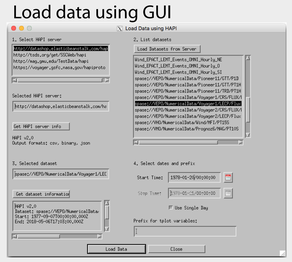


MIDL4
`http://cassini-mimi.jhuapl.edu/MIDL/HAPI`



SPEDAS
IDL tool suite that reads HAPI data
`http://spedas.org`

Load data using GUI



Very simple code to load HAPI data from any server.

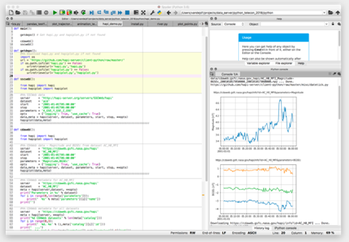
```
hapi_load_data, /catalog, server='http://datashop.elasticbeanstalk.com/hapi'

hapi_load_data, /info, dataset='LEMMS_PHA_CHANNELS_FULL_TIME_RES', $
        server='http://datashop.elasticbeanstalk.com/hapi'

hapi_load_data, trange=['04-07-01', '04-07-02'], $
        dataset='LEMMS_PHA_CHANNELS_FULL_TIME_RES', $
        server='http://datashop.elasticbeanstalk.com/hapi'

tplot, 'proton_spectrum'
```

Python Client
basic example for data retrieval
`https://github.com/hapi-server/client-python`



Demonstration code that makes simple plots. The point is not the plots, but the data loading capability.

The same IDL or Python code can access multiple missions!!

Next steps: create generic servers - drop-in server to expose existing data via HAPI
integrate Python client into emerging Heliophysics Python library
modify the spec to allow for time-varying spetral bins

CDAWeb
Goddard Space Flight Center
`https://cdaweb.gsfc.nasa.gov/hapi`

Integrated Space Weather Analysis System
`https://iswa.gsfc.nasa.gov/IswaSystemWebApp/hapi`

CDPP
Plasma Physics Data Center

European Space Astronomy Centre

PDS: Planetary Plasma Interactions Node
`https://pds-ppi.igpp.ucla.edu/hapi`

HAPI server up soon!

Space Exploration Sector
`http://datashop.elasticbeanstalk.com/hapi`

Department of Physics and Astronomy
`http://hapi-server.org/servers/SSCWeb/hapi`

RADIO AND PLASMA WAVE GROUP
`http://planet.physics.uiowa.edu/das/das2Server/hapi`