JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

HAPI and SPASE:

how to leverage interoperable discovery and access for easier science analysis

Jon Vandegriff, JHU / APL
Robert S. Weigel, George Mason University
Jeremy Faden, d.b.a. Cottage Systems

Outline

- What is HAPI?
  - getting everyone on the same page
- What capabilities are built to use the HAPI standard
  - servers that make data available
  - testing mechanisms that allow servers to be checked against the spec
  - client libraries that enable clients to use HAPI to get data
  - clients that have added their own HAPI reading capability
- Short demo of a web-based HAPI client: http://hapi-server.org/servers
- Integrating HAPI and SPASE:
  - SPASE to HAPI: can comprehensive SPASE support a drop-in HAPI server?
  - HAPI to SPASE: can a HAPI server be used to create SPASE records?
  - the importance of file listings and URI templates

Purpose of HAPI

The same as with every standard: to enhance interoperability!

What if every dataset you wanted to use was accessible through a single API?

HAPI aims to solve the "FILL MY ARRAY" problem

Scientist: "I don't care about where the data is, file formats, which files have which data, etc"
"I want data to just appear in my analysis code ready to use."

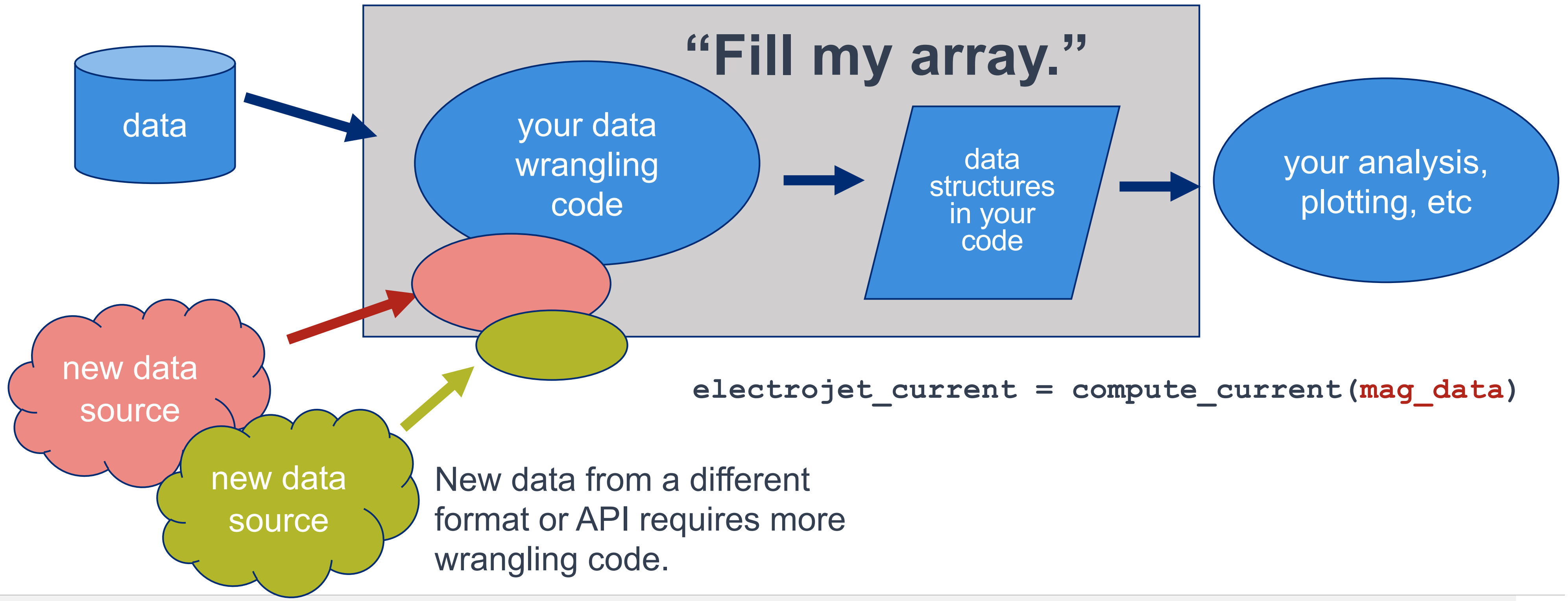
```
# some kind of read routine to obtain data:
mag_data = read_data( dataset, selection_criteria )

# the mag_data variable is a structure that you use
# and your analysis and plotting routines "recognize"
electrojet_current  =  compute_current(mag_data)
```

How the data gets into the `mag_data` array is the concern of HAPI infrastructure question.

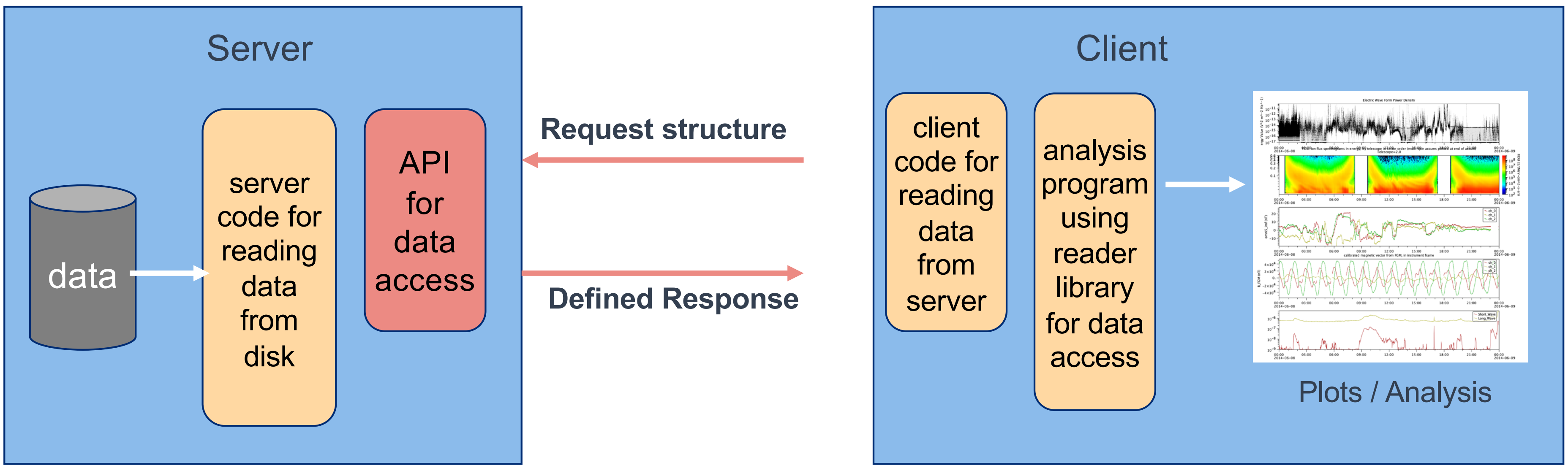
HAPI aims to solve the “FILL MY ARRAY” problem

Scientist: “I don’t care about where the data is, file formats, which files have which data, etc”
“I want data to just appear in my analysis code ready to use.”



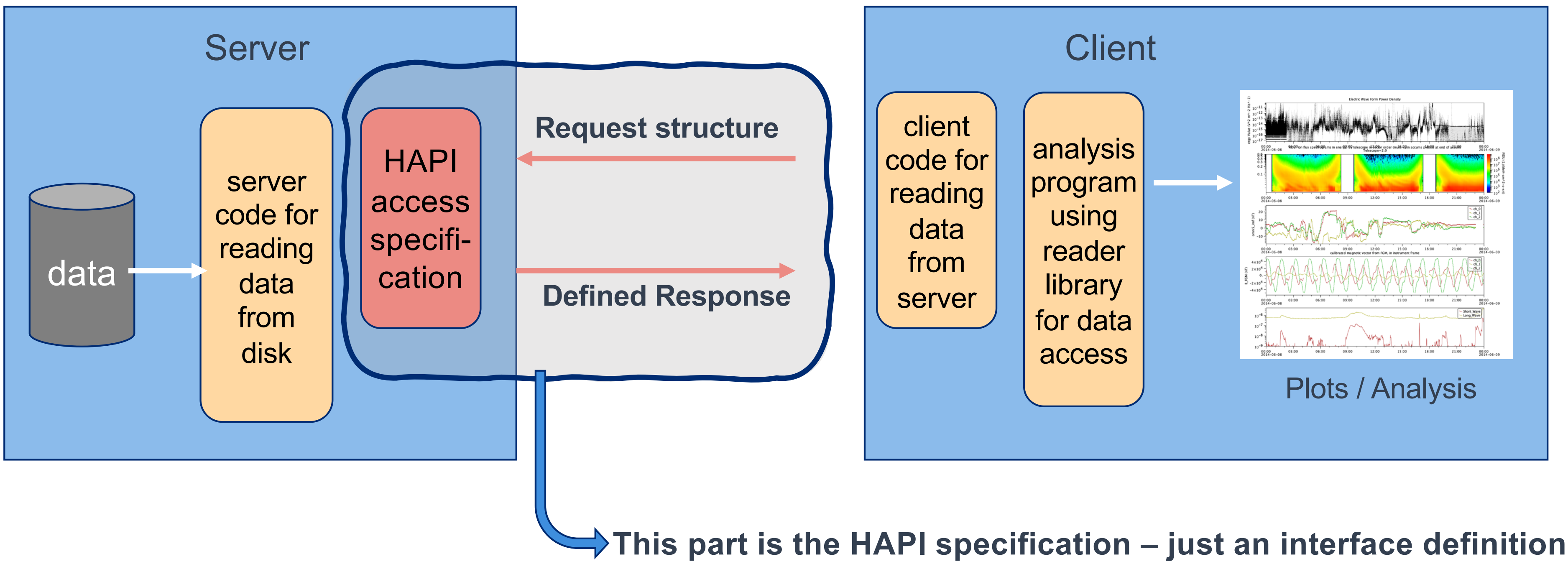
What is HAPI?

- HAPI = Heliophysics Application Programmer's Interface
- a standard interface for serving tine series data
- a way to achieve lasting interoperability for Heliophysics data analysis



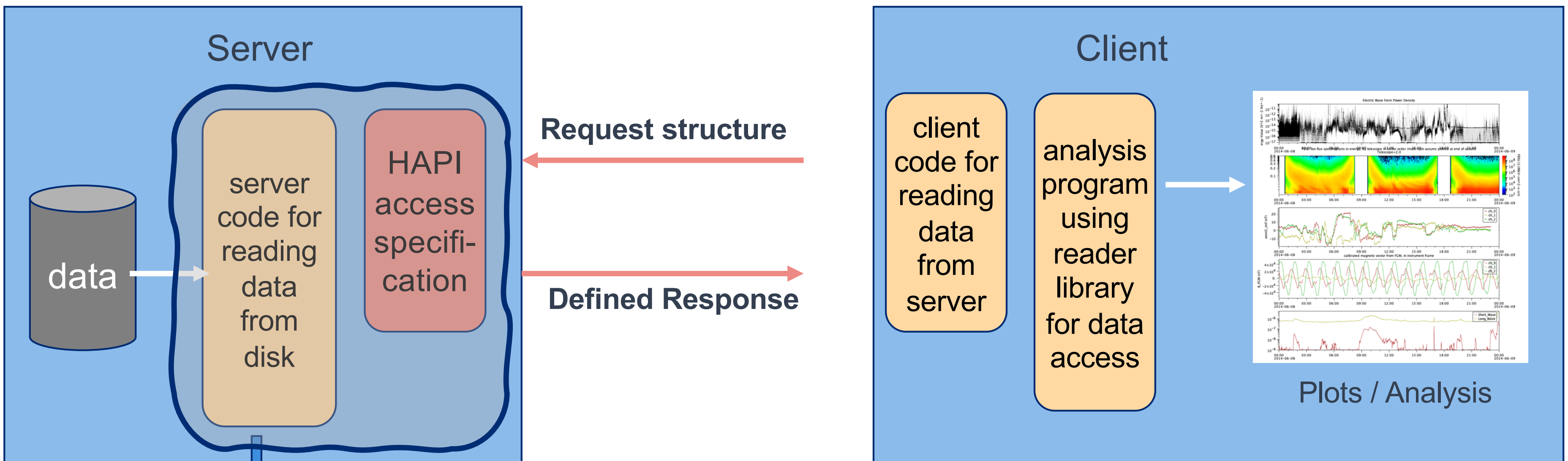
What is HAPI?

- HAPI = Heliophysics Application Programmer’s Interface
- a standard interface for serving tine series data
- a way to achieve lasting interoperability for Heliophysics data analysis



What is HAPI?

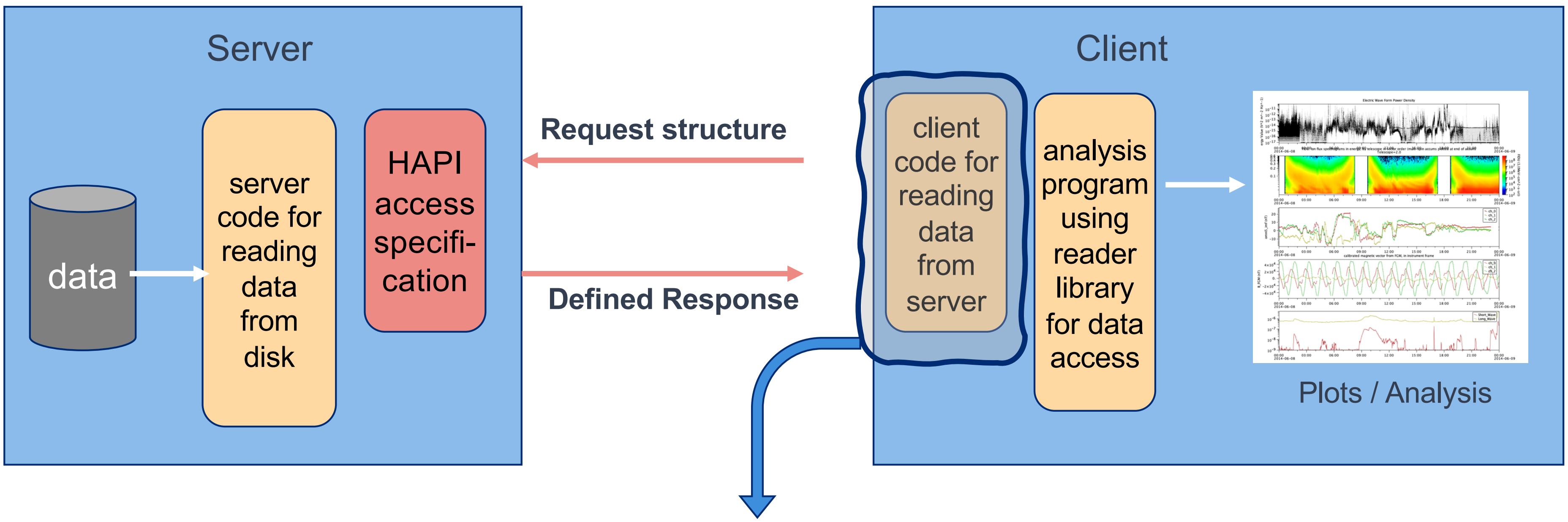
- HAPI = Heliophysics Application Programmer’s Interface
- a standard interface for serving tine series data
- a way to achieve lasting interoperability for Heliophysics data analysis



The code on the server that implements the spec to deliver data is a HAPI server.

What is HAPI?

- HAPI = Heliophysics Application Programmer’s Interface
- a standard interface for serving tine series data
- a way to achieve lasting interoperability for Heliophysics data analysis

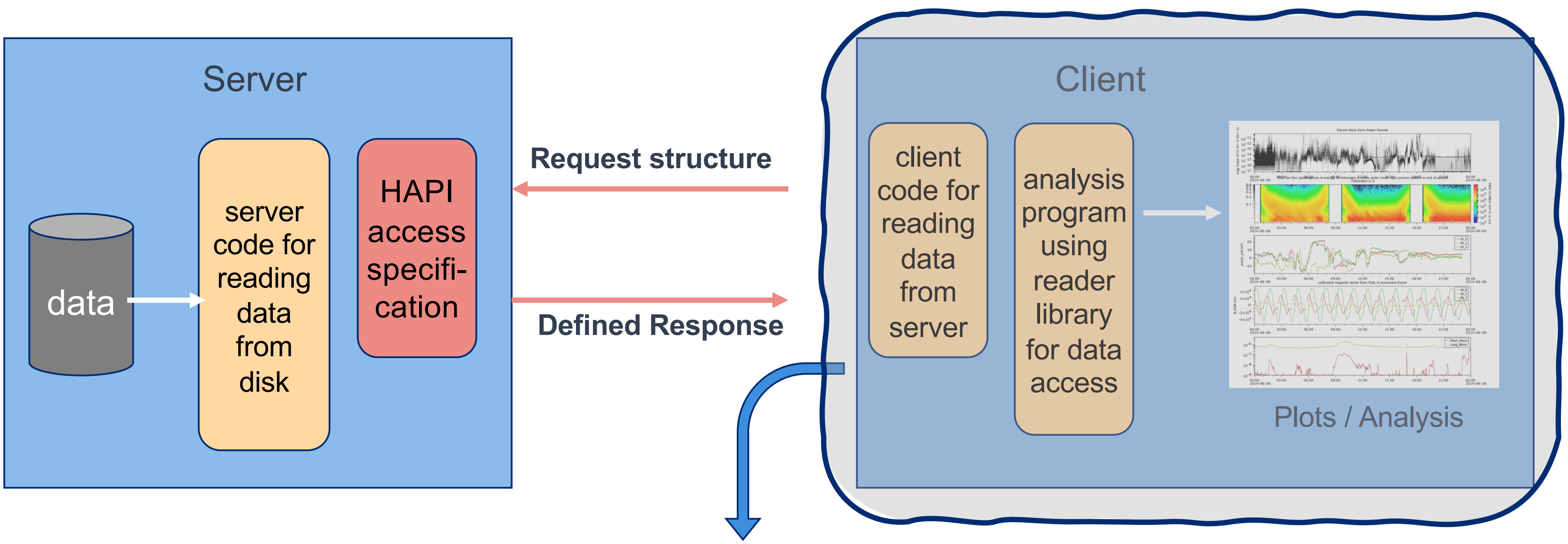


The library that interacts with a HAPI server is the HAPI client library.

PyHC core package => HAPI Python client library.

What is HAPI?

- HAPI = Heliophysics Application Programmer's Interface
- a standard interface for serving tine series data
- a way to achieve lasting interoperability for Heliophysics data analysis



The entire analysis program is then a HAPI-enabled client.

(there are more than just Python clients)

HAPI Spec Details

- Think of HAPI as "**http for data**" – a protocol for asking and getting something from a server
- RESTful – no state so that each request is independent
- URLs representing the requests can be thought of as (semi-)persistent identifiers
- Endpoints define the things you can ask of a HAPI server

elements required by the data endpoint protocol

data endpoint

Request:

```
http://server.org/hapi/data?dataset=ACE_MAG
                           &start=2004-183T00:00Z
                           &stop=2004-184T00:00Z
```

(line breaks for clarity)

Response:

```
2004-183T00:00:03.403Z,  1.0724e+02, -6.8993e+01, -5.1978e+02
2004-183T00:00:07.153Z,  1.0842e+02, -6.8956e+01, -5.1962e+02
2004-183T00:00:10.907Z,  1.0855e+02, -6.9063e+01, -5.2084e+02
2004-183T00:00:14.653Z,  1.0852e+02, -6.9049e+01, -5.2085e+02
2004-183T00:00:18.403Z,  1.0849e+02, -6.9035e+01, -5.2085e+02
2004-183T00:00:22.153Z,  1.0862e+02, -6.9142e+01, -5.2207e+02
2004-183T00:00:25.903Z,  1.0859e+02, -6.9128e+01, -5.2208e+02
```

(always the same format for all servers)

HAPI defines 5 URL endpoints every server must have

Endpoints must be directly below a URL that ends with ‘`hapi`’

- `http://example.com/`**`hapi`**`/`**`about`**
- `http://example.com/`**`hapi`**`/`**`capabilities`**
  - describes options implemented by the server
- `http://example.com/`**`hapi`**`/`**`catalog`**
  - list of datasets at the server
- `http://example.com/`**`hapi`**`/`**`info`**
  - show metadata for one dataset at a time (basically a data header)
- `http://example.com/`**`hapi`**`/`**`data`**
  - retrieve a stream of data content for one dataset over a specific time range

Note: The intent is for **computers** to read from these endpoints, but humans can look at them easily too (web browser, curl, etc)

Using HAPI to serve images

Time	SCAttitude	Quality	PixelLookDirns	URI string
t0	matrix[3,2]	q0	m0[500,500]	http://data.org/image1.fits
t1	matrix[3,2]	q1	m1[500,500]	http://data.org/image2.fits
t1	matrix[3,2]	q2	m2[500,500]	http://data.org/image3.fits
t2	matrix[3,2]	q3	m3[500,500]	http://data.org/image4.fits

- Images listed by reference as URI's (most likely URLs)
- many client programs know what to do with image data (standards are old / stable)
- each image's metadata can be listed in other HAPI parameters in each record
- filtering of images by metadata properties can be done on the client using client specific knowledge about this image type (another standard here could be useful)

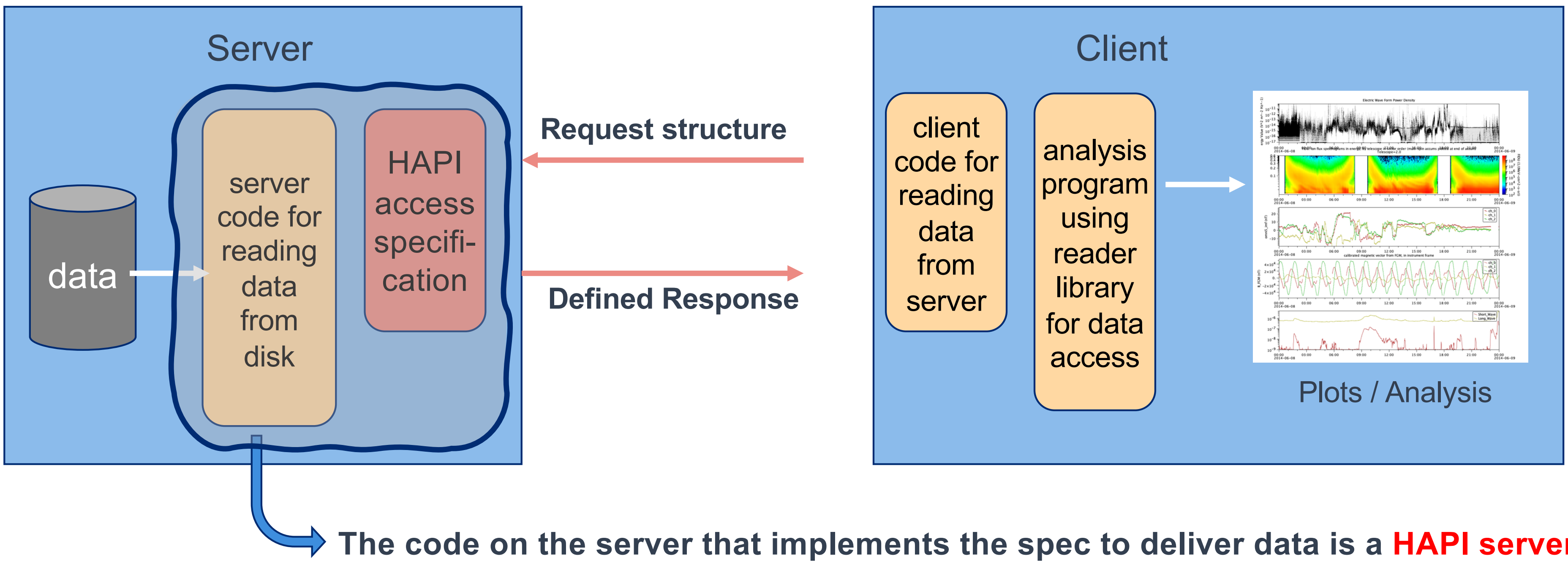
- DANGER: do not use this to list time series files (then you are just a listing service)

Outline

- What is HAPI?
  - getting everyone on the same page
- **What capabilities are built to use the HAPI standard**
  - **servers that make data available**
  - **testing mechanisms that allow servers to be checked against the spec**
  - **client libraries that enable clients to use HAPI to get data**
  - **clients that have added their own HAPI reading capability**
- Short demo of a web-based HAPI client: http://hapi-server.org/servers
- Integrating HAPI and SPASE:
  - SPASE to HAPI: can comprehensive SPASE support a drop-in HAPI server?
  - HAPI to SAPSE: can a HAPI server be used to create SPASE records?
  - the importance of file listings and URI templates

What is HAPI?

- HAPI = Heliophysics Application Programmer's Interface
- a standard interface for serving tine series data
- a way to achieve lasting interoperability for Heliophysics data analysis



The code on the server that implements the spec to deliver data is a HAPI server.

Servers that meet the HAPI SPEC

- CDAWeb – Java server written by Nand Lal; being re-written by Jeremy and Bob
- SSCWeb – there is a pass-through server written by Bob (reads from SSCWeb and reformats results to be HAPI compliant)
- CCMC – ISWAT space weather indices and parameters; custom server they wrote themselves
- LISIRD server at LASP – solar irradiance data; added HAPI output mechanism to already modular server (was easy to change output modality to support HAPI)
- AMDA – French server of planetary and Heliophysics data; custom implementation; has multiple output modes, including the more complex Das2 stream (from Autoplot)
- SWARM mission – has some data in HAPI; custom implementation
- KNLM – Dutch Meteorological Society; custom implementation
- PDS – PPI node of Planetary Data System – custom implementation in Java
- ESAC – European Space Astronomy Center in Madrid; generic Java server with custom help from HAPI team; about to be online
- SuperMAG – about to be online; mods to existing server
- Also hoping to work with MADRIGAL database and the CEDAR community

What data is available now with HAPI?

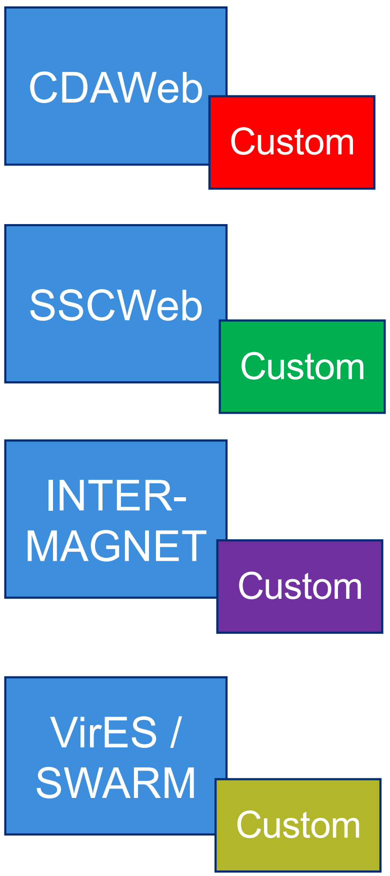
Institution	Server	Type of Data	Number of Datasets
	CDAWeb	Heliophysics	2800
	SSCWeb	Ephemeris	250
	CCMC	Space Weather	250
	AMDA	Heliophysics and Planetary Data and Ephemeris	500
University of Iowa	Das2 Server	Helio. & Planetary	30
	LISIRD	Solar Irradiance	40
SWARM Mission	ViRES Data Server	Space Mag Data	14
INTERMAGNET	INTERMAGNET	Ground-based Mag	~1000

Note: Both European and US buy-in.

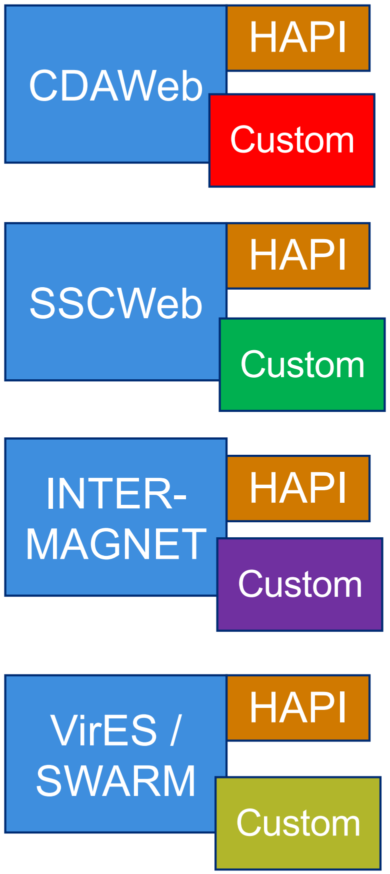
Coming soon

Institution	Server	Type of Data	Number of Datasets
	ESAC	Heliophysics	2073
JHU / APL	SuperMAG	global ground mag	~500
JHU / APL	TIMED / GUVI	ionospheric images	~10
	PDS PPI Node	Planetary Plasma, Particle, and Fields	~1000
Royal Netherlands Meteorological Institute.	KNMI	Space Weather	~100

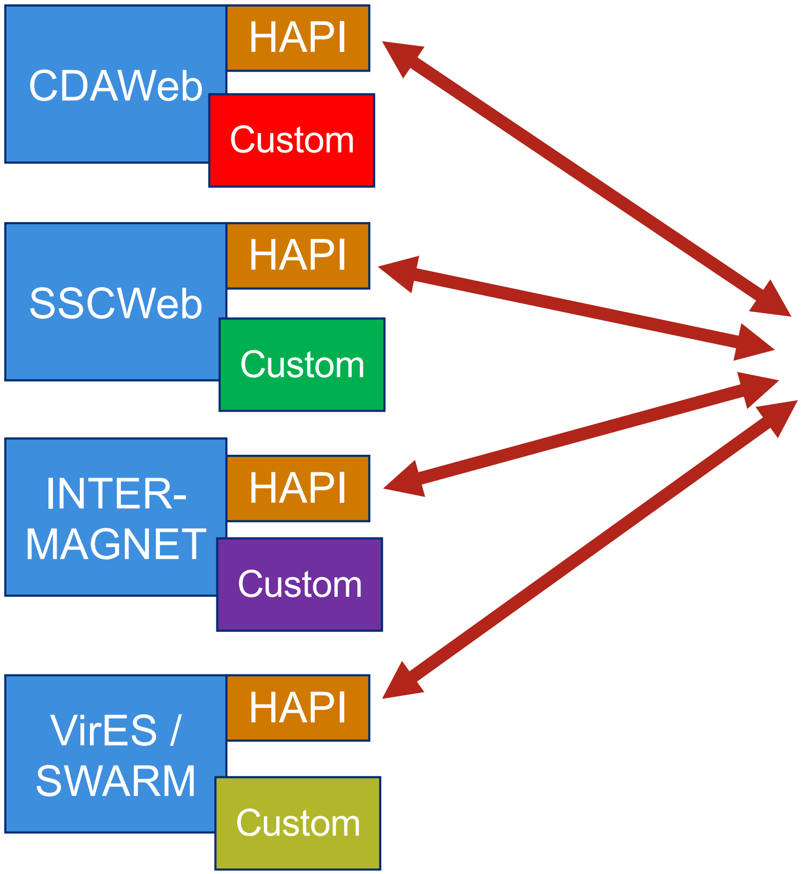
Eventually			
CSA	Space Environment Canada (new initiative)	Ground-based ionospheric data	~1000
CEDAR / NSF	Madrigal	Space Weather	1000+ (??)



Before HAPI, each service already had its own computer-based access mechanism (i.e., it’s own API).



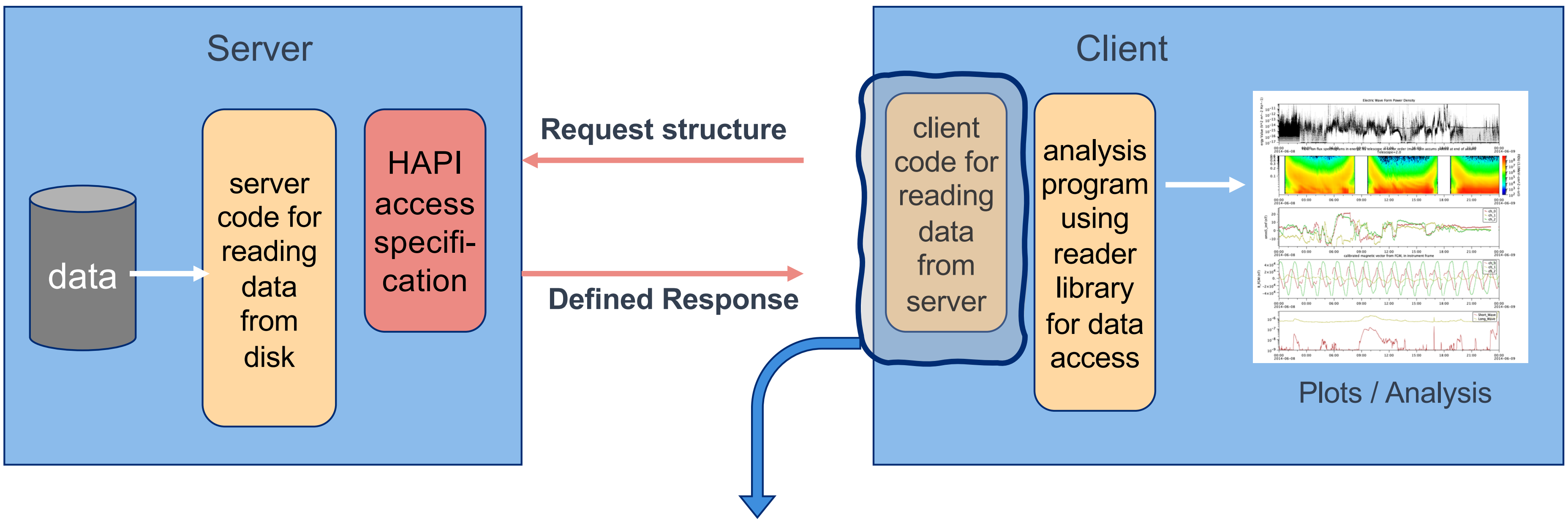
Now, each service has also ADDED another way to access data using HAPI.



So the content from each of these services can be reached with a single piece of code.

This brings us to the topic of
HAPI clients

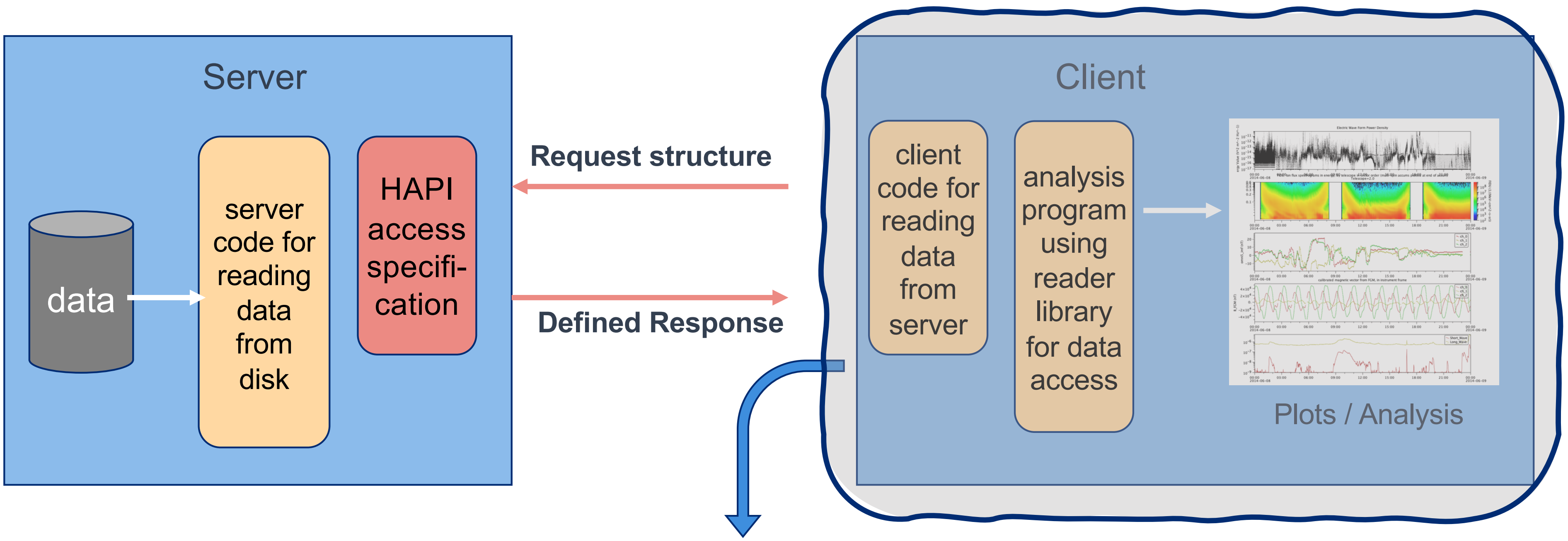
Recall that a HAPI client can be just the HAPI-specific library that knows how to talk to a HAPI server, or…



The library that interacts with a HAPI server is the HAPI client library.

PyHC core package => HAPI Python client library.

A HAPI client can be a full-fledged application that uses it’s own code or a HAPI client library to talk to a HAPI server.



The entire analysis program is then a HAPI-enabled client.

(there are more than just Python clients)

HAPI Clients

Libraries

- Python library (Bob Weigel)
- IDL library (Scott Boardsen)
- Java library (Jeremy Faden, Larry Brown)
- Matllab library (Bob Weigel)
- R library (Daniel Wilborn)

Applications

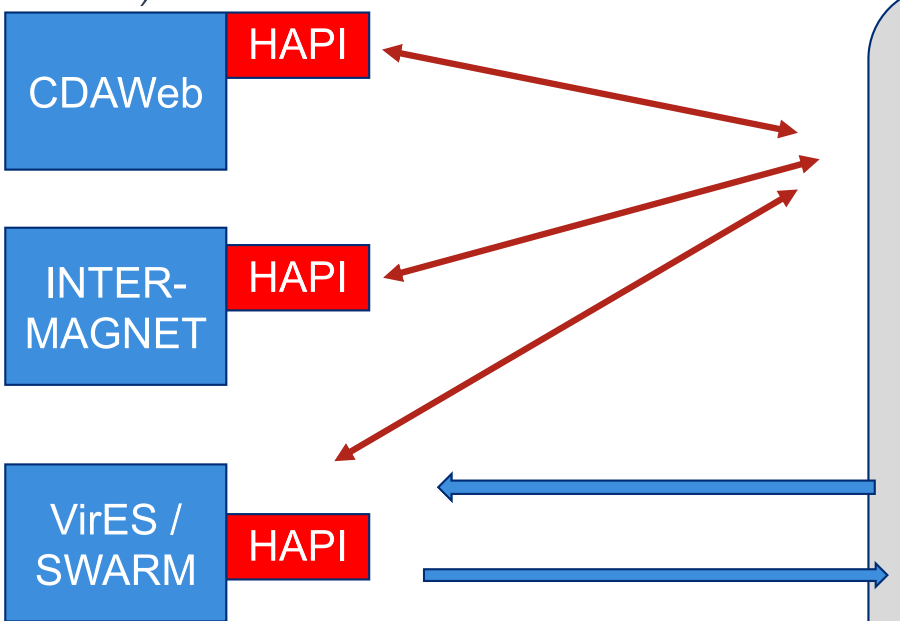
- SPEDAS (IDL, Eric Grimes)
- PySPEDAS (Eric Grimes; uses above Python library)
- Autoplot (Jeremy Faden – uses internal code)
- hapi-server.org/servers (JavaScript, Bob Weigel)
- KNLM visualizer (JavaScript, Eelco Doornbos)
- LASP Space Weather Data Portal (JavaScript, Jenny Knuth)

Outline

- What is HAPI?
  - getting everyone on the same page
- What capabilities are built to use the HAPI standard
  - servers that make data available
  - testing mechanisms that allow servers to be checked against the spec
  - client libraries that enable clients to use HAPI to get data
  - clients that have added their own HAPI reading capability
- **Short demo of a web-based HAPI client: http://hapi-server.org/servers**
- Integrating HAPI and SPASE:
  - SPASE to HAPI: can comprehensive SPASE support a drop-in HAPI server?
  - HAPI to SAPSE: can a HAPI server be used to create SPASE records?
  - the importance of file listings and URI templates

Trying HAPI Access – audience participation

Some HAPI Servers
(not full list)



HAPI Client for demo purposes

`http://hapi-server.org/servers`

JavaScript page -- uses public list of known servers and offers a simple way to explore data at each server

constructs HAPI query to a server

offers a simple view of the returned data (raw numbers or a basic plot)

coding assistance too!

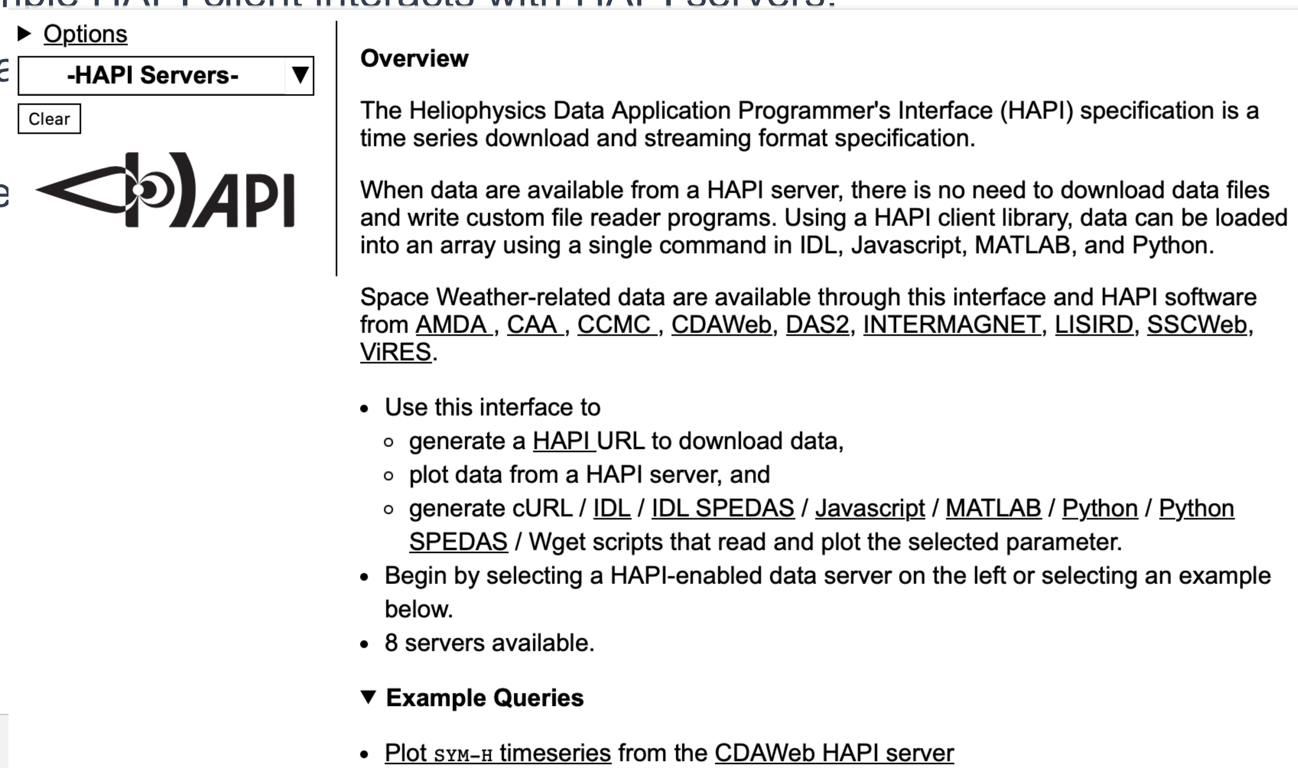
HAPI Access Illustrated using a JavaScript client

Go to this site: `http://hapi-server.org/servers`

Explore how a simple HAPI client interacts with HAPI servers.

This heavily huma … out it
allows you to see

It can create code



▶ Options

-HAPI Servers-

Clear

Overview

The Heliophysics Data Application Programmer's Interface (HAPI) specification is a time series download and streaming format specification.

When data are available from a HAPI server, there is no need to download data files and write custom file reader programs. Using a HAPI client library, data can be loaded into an array using a single command in IDL, Javascript, MATLAB, and Python.

Space Weather-related data are available through this interface and HAPI software from AMDA , CAA , CCMC , CDAWeb, DAS2, INTERMAGNET, LISIRD, SSCWeb, ViRES.

- Use this interface to
  - generate a HAPI URL to download data,
  - plot data from a HAPI server, and
  - generate cURL / IDL / IDL SPEDAS / Javascript / MATLAB / Python / Python SPEDAS / Wget scripts that read and plot the selected parameter.
- Begin by selecting a HAPI-enabled data server on the left or selecting an example below.
- 8 servers available.

▼ **Example Queries**

- Plot SYM-H timeseries from the CDAWeb HAPI server

Try out HAPI – go now to: `http://hapi-server.org/servers`

Steps to take:
For "HAPI servers" : select CDAWeb
For "Datasets": select RBSP-A_DENSITY_EMFISIS-L4
For "Parameters": select "electron density"
For the time range: the defaults are ok
For output: select Data
For format: select CSV

Other options:

For output: select Image
For format: select PNG

For output: select Script
For format: select Python

What to notice:
HAPI URL is shown to you (see below)
Data is returned to the screen, but normally
this would appear in your code (FILL MY ARRAY)

https://cdaweb.gsfc.nasa.gov/hapi/data?id=RBSP-A_DENSITY_EMFISIS-L4&parameters=density&time.min=2012-09-15T00:00:03Z&time.max=2012-09-17T00:00:03.000Z

Python code stub generated for you

```python
# example showing how to get OMNIWeb data
from hapiclient import hapi

server     = 'https://cdaweb.gsfc.nasa.gov/hapi'
dataset    = 'OMNI2_H0_MRG1HR'
start      = '2021-10-25T00:00:00Z'
stop       = '2021-12-01T00:00:00Z'
# parameters is a comma-separated list
parameters = 'DST1800,Proton_QI1800'

# Configuration options for the hapi function.
opts = {'logging': True, 'usecache': True, 'cachedir': './hapicache' }

# Get parameter data
data, meta = hapi(server, dataset, parameters, start, stop, **opts)
```

Outline

- What is HAPI?
  - getting everyone on the same page
- What capabilities are built to use the HAPI standard
  - servers that make data available
  - testing mechanisms that allow servers to be checked against the spec
  - client libraries that enable clients to use HAPI to get data
  - clients that have added their own HAPI reading capability
- Short demo of a web-based HAPI client: http://hapi-server.org/servers
- Integrating HAPI and SPASE:
  - SPASE to HAPI: can comprehensive SPASE support a drop-in HAPI server?
  - HAPI to SAPSE: can a HAPI server be used to create SPASE records?
  - the importance of file listings and URI templates

HAPI is for data access; SPASE is for dataset discovery

Scientist: "I don't care about where the data is, file formats, which files have which data, etc"
"I want data to just appear in my analysis code ready to use."

SPASE can help you determine which dataset to use

```
# some kind of read routine to obtain data:
mag_data = read_data( dataset, selection_criteria )

# the mag_data variable is a structure that you use
# and your analysis and plotting routines "recognize"
electrojet_current  =  compute_current(mag_data)
```

How the data gets into the `mag_data` array is the concern of HAPI infrastructure question.

One options for Integration: SPASE-driven HAPI server

- If you have these four things:
  1. well-formed SPASE that maps one-to-one with data file contents
  2. a self-describing data format (CDF, netCDF, HDF); (not CSV, for example)
  3. well-formed file names that make it easy to tell which date ranges of data are in each file
  4. open data access: no access restrictions like username, password, login, etc
- Then it would be easy to make a generic HAPI server that:
  1. uses the SPASE to flesh out all required HAPI metadata
  2. uses the self-described format to read data into the server to the data
  3. handles all HAPI request and response details

Constraints for using SPASE to drive a HAPI Server

1. The order of parameters in the SPASE record would need to match that used in the HAPI server (or a look-up table provided)
2. All of the parameters in the dataset would need to have the same timestamps (or a look-up table provided) (some SPASE records refer to contents with more than one DEPEND0 for time, and different cadences)
3. Metadata for bins (DEPEND_2 in CDF – things like spectral energy or frequency ranges, etc), would need to be in the SPASE records
4. Ideally, the start/stop dates would be updated on a regular basis for datasets that are still growing
5. All parameters in the SPASE record would be available from the HAPI server (or a look-up table provided to indicate which ones are not available) (example: time_pb5 – this would need to be masked out for HAPI if you are using a pass-through – a native CDF read would find this variable; time_pb5 is old, super-ceded time scheme; depends on var_type for CDFs and maybe display_type)

Most of the time, all 5 of these are not guaranteed, so making a SPASE-driven HAPI server is not practical.

About five years ago, we looked into using the CDAWeb SPASE records to create a server (Bernie has a demo server running). At that time, about 20-30% of CDAWeb had SPASE records. At present, the fraction is about 60% or so. Given this coverage issue, so much help is needed from metadata in the master CDFs, and HAPI metadata can be created based on metadata in master CDFs, for the new CDAWeb HAPI server, we are primarily master CDF driven. The exception is that we use units in the SPASE record because they seem to follow a convention. Does SPASE follow a machine-parsable schema for unit strings so that it is caught when there is an error?

The other part of SPASE we are considering using is the coordinate system name and labels for vector components when available.

Example of parameter ordering issue

- SPASE lists parameters: a, b, c
- If CDF has them in this order: b, c, a
  - then a HAPI server would possibly get confused by the SPASE

Overlapping need: Coordinate System Standardization

- The other part of SPASE that the HAPI team is considering using is the coordinate system name and labels for vector components when available.
- HAPI allows specification of computer readable coord. sys names – only SPASE has this for Heliophysics (as far as we can tell)
- See next slide for screen-snap of SPASE coordinate system names. These are not really formal definitions, but the names are unique and publicly accessible by a computer, given, and there is no other place to get this kind of coordinate naming info (for Heliophysics) where else that is doing this right now.

Element spase:CoordinateSystem / spase:CoordinateSystemName

Namespace	http://www.spase-group.org/data/schema		
Diagram			
Type	**spase:CoordinateSystemName**		
Properties	Content	**simple**	
	MinOccurs	**1**	
	MaxOccurs	**1**	
Facets	Enumeration	**Carrington**	A coordinate system which is centered at the Sun and is "fixed" with respect to the synodic rotation rate; the mean sync days. The Astronomical Almanac gives a value for Carrington longitude of 349.03 degrees at 0000 UT on 1 January 1995.
	Enumeration	**CGM**	Corrected Geomagnetic - A coordinate system from a spatial point with GEO radial distance and geomagnetic latitude and l appropriate IGRF/DGRF model field vector through to the point where the field line crosses the geomagnetic dipole equator dipole magnetic field vector Earthward from that point on the equatorial plane, in the same hemisphere as the original point distance is reached. Designate the dipole latitude and longitude at that point as the CGM latitude and longitude of the origin <http://nssdc.gsfc.nasa.gov/space/cgm/cgmm_des.html>
	Enumeration	**CSO**	Corrected Solar Orbital - A coordinate system related to Earth where X is anti-sunward, Y along the orbital velocity direction
	Enumeration	**DM**	Dipole Meridian - A coordinate system centered at the observation point. Z axis is parallel to the Earths dipole axis, positive plane defined by Z and the line linking the observation point with the Earths center. Y is positive eastward. See <http://cdpp
	Enumeration	**ECD**	Eccentric Dipole (ECD) coordinate system that aligns with a dipole whose origin and orientation may be different from the p of the containing body. The IGRF-12 coefficients for 2015 are used to determine the origin for the earth. The 2015 positions latitude: 86.29, longitude -160.06. South dip pole latitude: -64.28, longitude: 136.59, North geometric pole latitude: 80.37, lor geomagnetic pole latitude: -80.37, longitude: 107.37ECD is defined in doi:10.1186/s40623-015-0228-9.
	Enumeration	**ECEF**	The Earth-Centered, Earth-Fixed (ECEF) coordinate system has point (0,0,0) defined as the center of mass of the Earth. Its International Reference Pole (IRP) and International Reference Meridian (IRM). The x-axis intersects the sphere of the Eart (Equator) and 0 degree longitude (Greenwich). The z-axis points north. The y-axis completes the right handed coordinate s
	Enumeration	**ENP**	ENP (also called PEN) - The P vector component points northward, perpendicular to orbit plane which for a zero degree inc Earths spin axis. The E vector component is perpendicular to P and N and points earthward. The N component is perpendic positive eastward.
	Enumeration	**GEI**	GEI Geocentric Equatorial Inertial - A coordinate system where the Z axis is along Earths spin vector, positive northward. X first point of Aries (from the Earth towards the Sun at the vernal equinox). See Russell, 1971. When the X axis is the directi equinox of J2000, the coordinate system is also called GCI. Then the Z axis is also defined as being normal to the mean Ea
	Enumeration	**GEO**	Geographic - geocentric corotating - A coordinate system where the Z axis is along Earths spin vector, positive northward. X meridian, positive towards Greenwich. See Russell, 1971.
	Enumeration	**GPHIO**	Kronian Solar Orbital - A coordinate system related to Saturn where X is anti-sunward, Y along the orbital velocity direction.
	Enumeration	**GSE**	Geocentric Solar Ecliptic - A coordinate system where the X axis is from Earth to Sun. Z axis is normal to the ecliptic, positiv 1971.
	Enumeration	**GSEQ**	Geocentric Solar Equatorial - A coordinate system where the X axis is from Earth to Sun. Y axis is parallel to solar equatoria northward. See Russell, 1971
	Enumeration	**GSM**	Geocentric Solar Magnetospheric - A coordinate system where the X axis is from Earth to Sun, Z axis is northward in a plan the geomagnetic dipole axis. See Russell, 1971
	Enumeration	**HAE**	Heliocentric Aries Ecliptic - A coordinate system where the Z axis is normal to the ecliptic plane, positive northward. X axis i

Other option for Integration: Harvesting SPASE from HAPI servers

- If your HAPI server has enough metadata, it would be possible to write code to grab the HAPI metadata and form it into a SPASE record
- There are communities that will be using HAPI that do not and will not be making their own SPASE metadata.
- It would be useful to us to capture their metadata into our own (emerging) federated system.

The importance of filenames (and URI templates)

- a URI template is a string representing a filename as a URL-like entity with date related variables:

- URL like: http://server.org/data/2023/mag_data_2023-01-01.cdf
- become this: **http://server.org/data/$Y/mag_data_$Y-$m-$d.cdf**

- Store the URI template strings for a dataset in the SPASE record to allow for easier, automated access to the hierarchy of files.