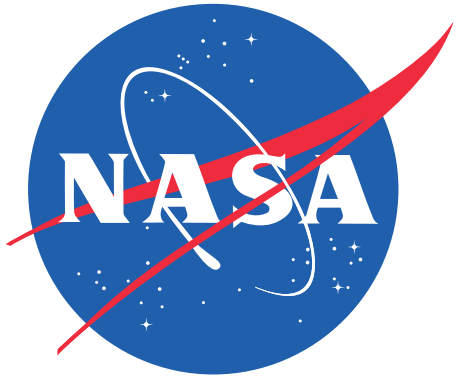
JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

Leveraging the HAPI Standard for Access to Time Series Data to Reduce the Data Access Burden on Scientists

Jon Vandegriff, JHU / APL
Robert S. Weigel, George Mason University
Jeremy Faden, Cottage Systems
Sandy Antunes, JHU / APL
Robert Candey (GSFC)
Doug Lindholm (LASP)
Bernard Harris (GSFC)



Royal Dutch Meteorological Institute

INTERMAGNET

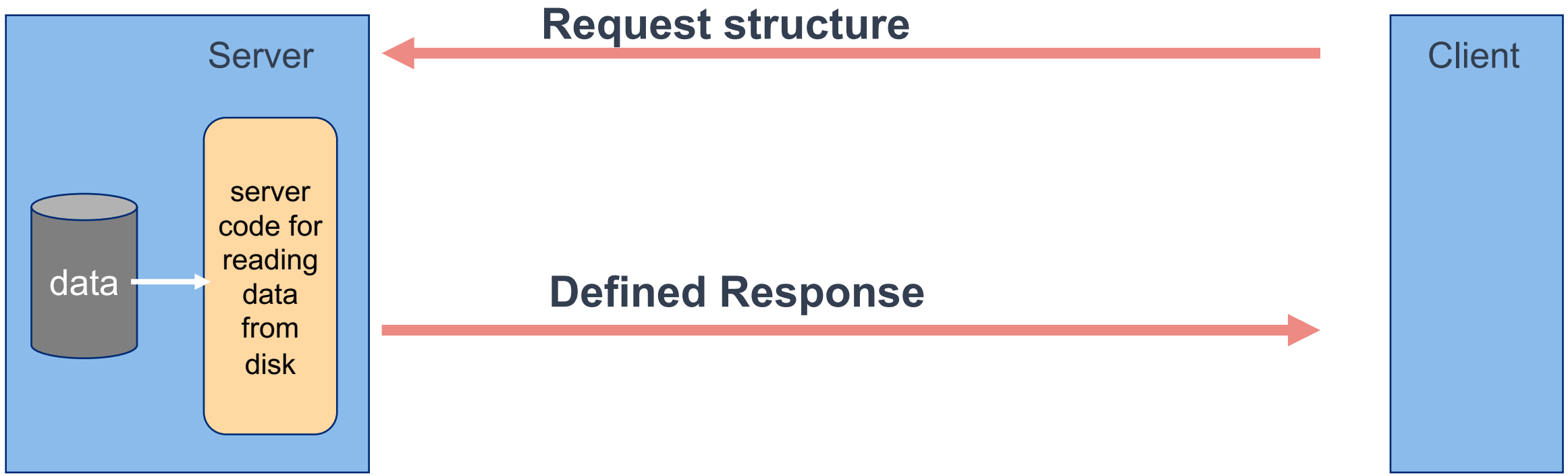
(many different types of organizations involved)

Outline

- What is HAPI?
- What are the benefits of using HAPI?
  - for scientists
  - for data centers
  - what about long-term?
- Demonstrations:
  - browse what is available with HAPI: http://hapi-server.org/servers
  - third-party client
  - (maybe) Python-based data access illustrated in a Jupyter notebook
- Q&A

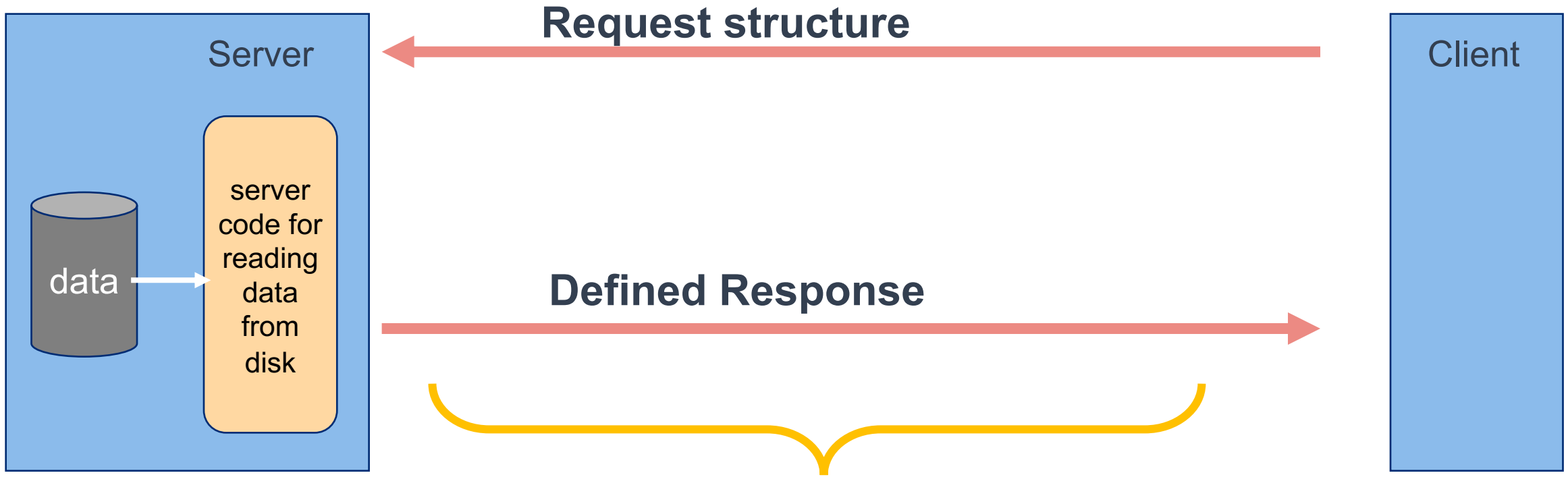
What is the Heliophysics Application Programmer's Interface (HAPI)?

A standard **interface** for serving time series data.



What is the Heliophysics Application Programmer's Interface (HAPI)?

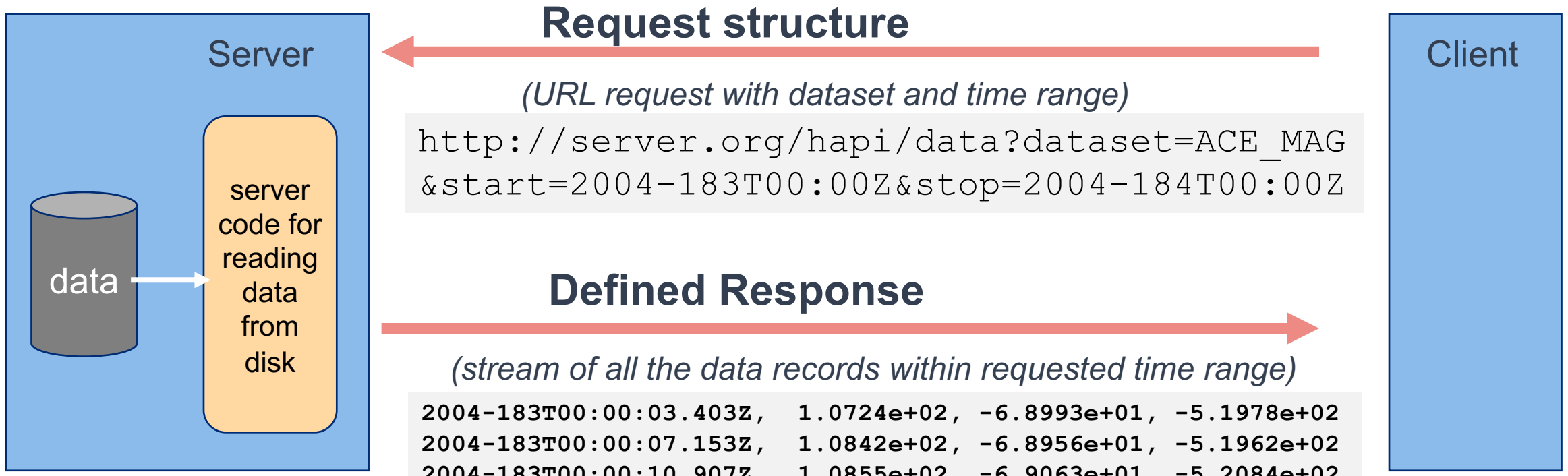
A standard **interface** for serving time series data.



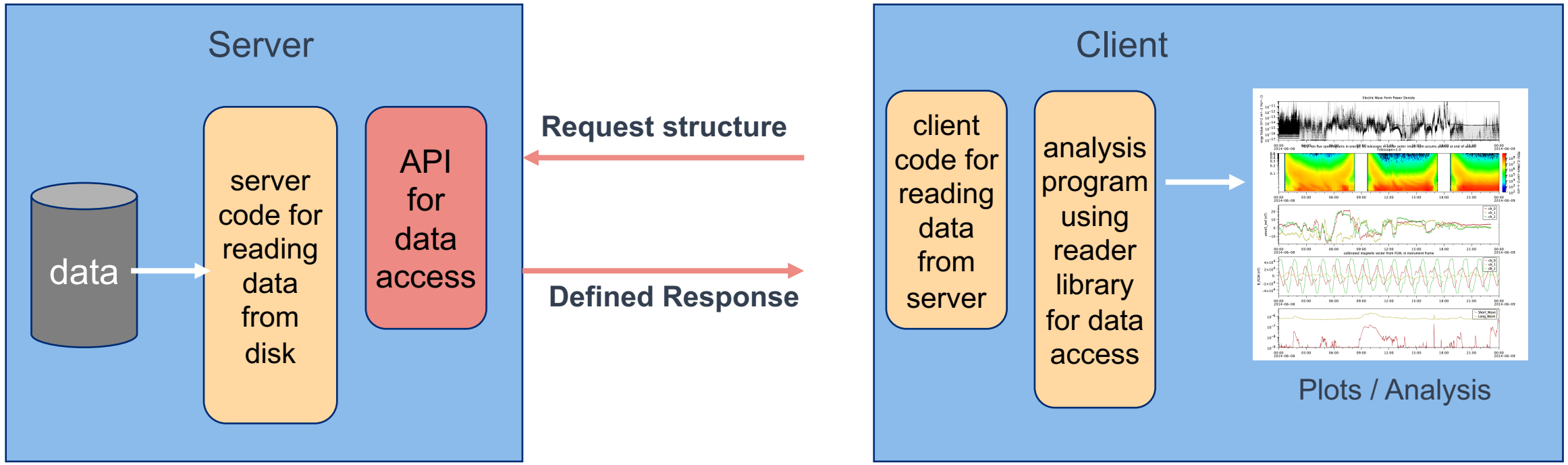
HAPI is a specification for this request / response exchange

What is the Heliophysics Application Programmer's Interface (HAPI)?

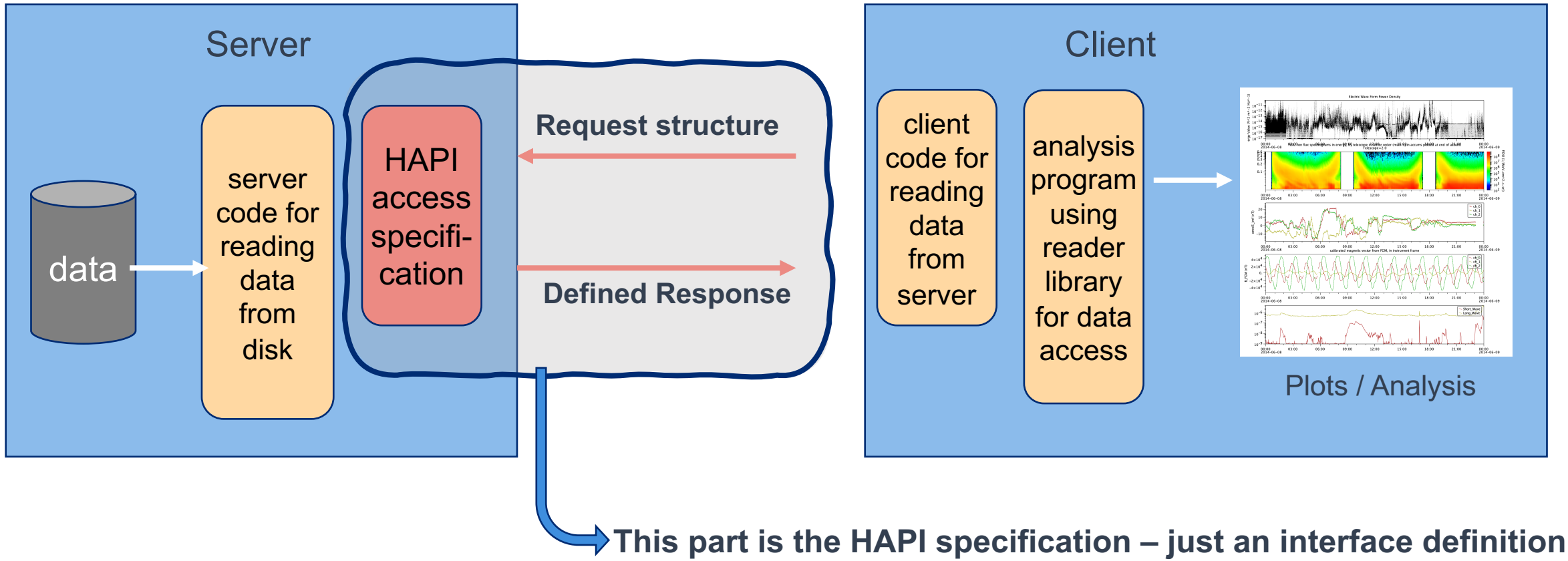
A standard **interface** for serving time series data.



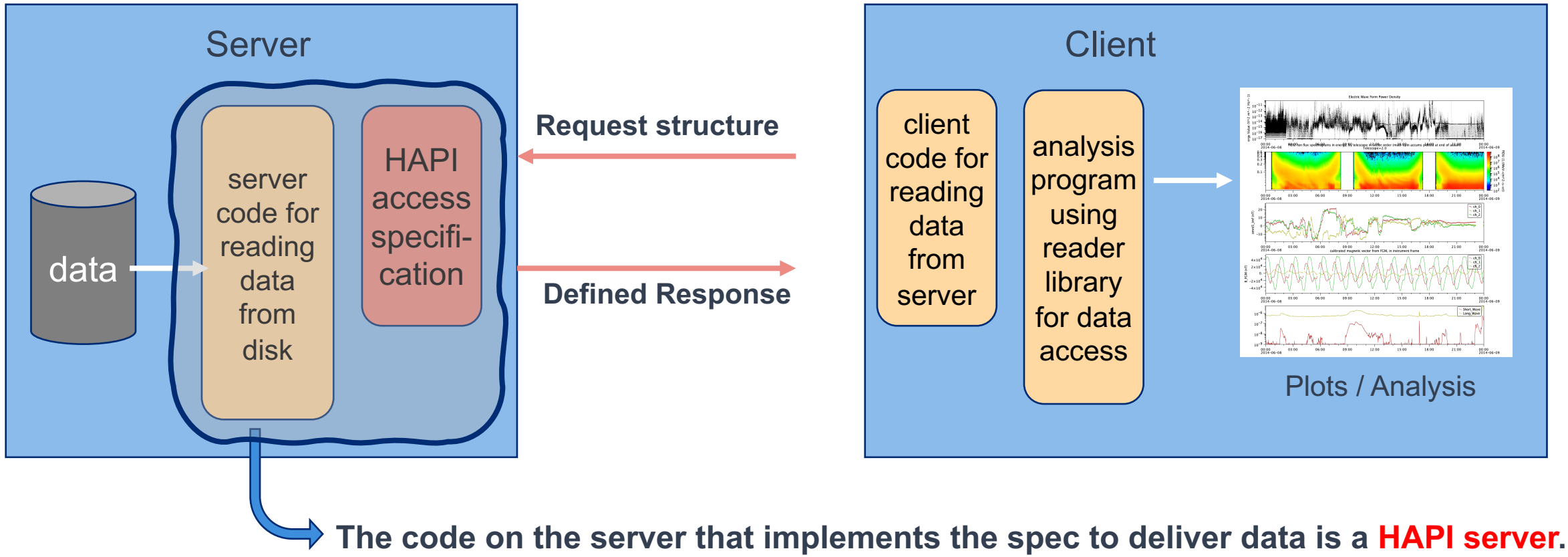
What is HAPI?


Server
data
server code for reading data from disk
API for data access
Request structure
Defined Response
Client
client code for reading data from server
analysis program using reader library for data access
Plots / Analysis

What is HAPI?

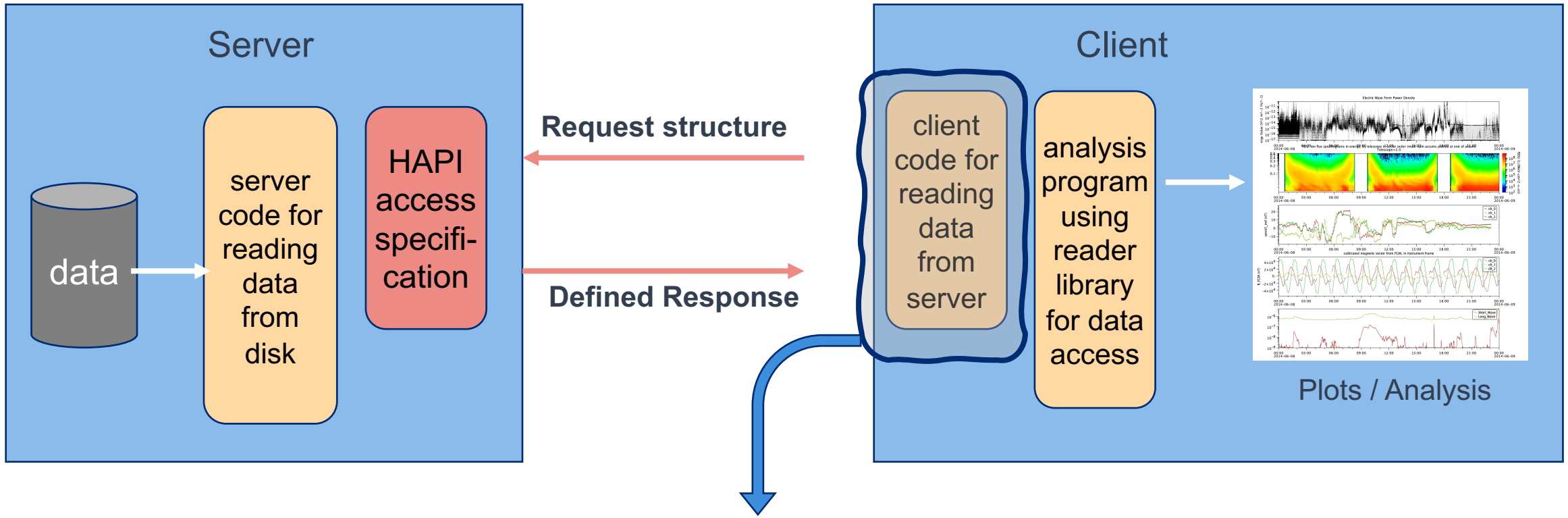

Server
data
server code for reading data from disk
HAPI access specifi-cation
Request structure
Defined Response
Client
client code for reading data from server
analysis program using reader library for data access
Plots / Analysis
This part is the HAPI specification – just an interface definition.

What is HAPI?



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

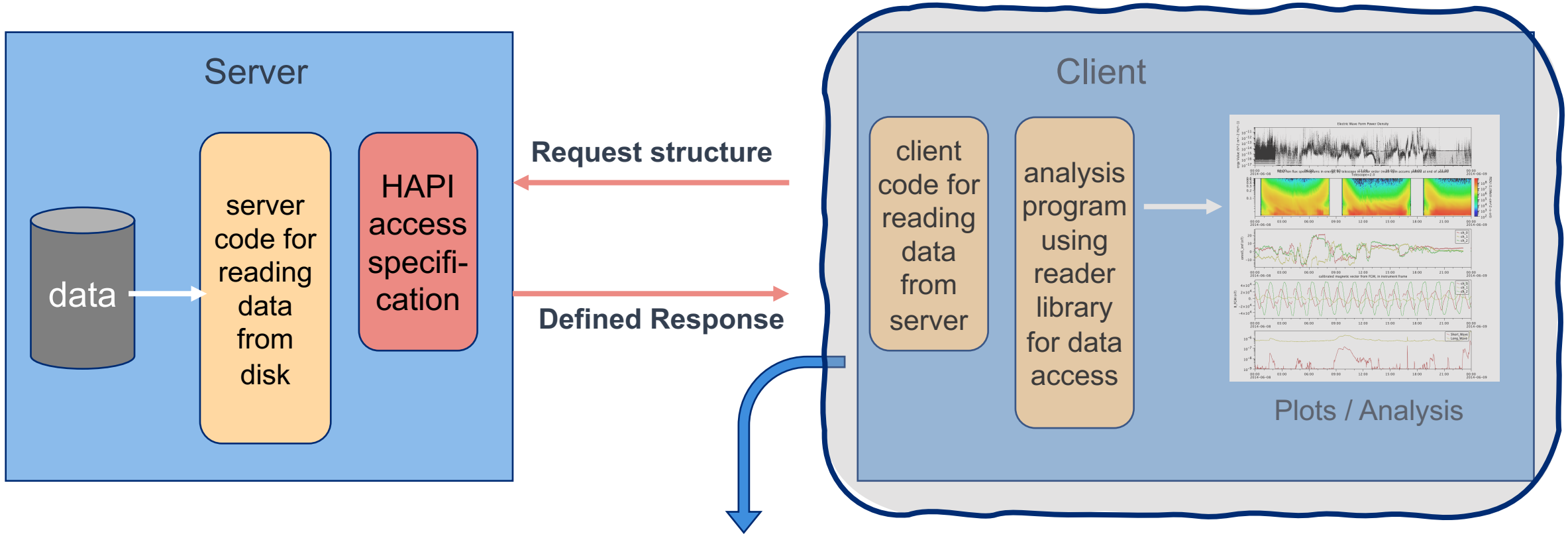
What is HAPI?



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

PyHC core package => HAPI Python client library.

What is HAPI?



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

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 Metadata is JSON

(just showing you that HAPI has good metadata - ask about in Q&A for details)


```
{
  "HAPI": "2.0", "status": {"code": 1200, "message": "OK"},
  "startDate": "1997-09-02T00:00:12Z",
  "stopDate": "2023-10-13T23:59:54Z",
  "resourceURL": "https://cdaweb.gsfc.nasa.gov/misc/NotesA.html#AC_H0_MFI",
  "contact": "N. Ness @ Bartol Research Institute"
  "parameters": [
    {"name": "Time", "type": "isotime", "units": "UTC", "length":24, "fill": null },
    {"name": "Magnitude", "type": "double", "units": "nT", "fill": "-1.0E31",
        "description": "B-field magnitude" },
    {"name": "BGSEc", "type": "double", "units": "nT", "fill": "-1.0E31", "size": [3],
        "description": "Magnetic Field Vector in GSE Cartesian coords (16 sec)",},
    {"name": "BGSM", "type": "double", "units": "nT", "fill": "-1.0E31", "size": [3],
        "description": "Magnetic field vector in GSM coordinates (16 sec)" },
    {"name": "dBrms", "type": "double", "units": "nT", "fill": "-1.0E31",
        "description": "RMS of Magnetic Field (16 sec period)" },
    {"name": "SC_pos_GSE", "type": "double", "units":"km", "fill": "-1.0E31",
        "size": [3], "description": "ACE s/c position, 3 comp. in GSE coord." },
    { "name": "SC_pos_GSM", "type": "double", "units": "km", "fill": "-1.0E31",
        "size": [3], "description": "ACE s/c position, 3 comp. in GSM coord." }  ]
}
```

HAPI Metadata

Key point: Just enough detail to be able to make an interpretable science plot.

-parameter names
-units
-fill value
-ability to describe 2D data (spectral data with energy or frequency bins)

HAPI defines 5 URL endpoints every server must have

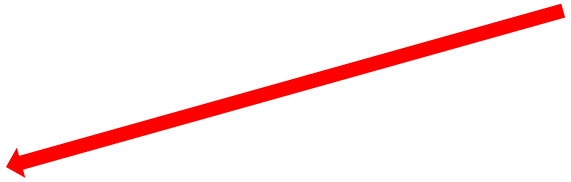
Endpoints must be directly below a URL that ends with '**`hapi`**'

- `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

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



Retrieve metadata for dataset

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

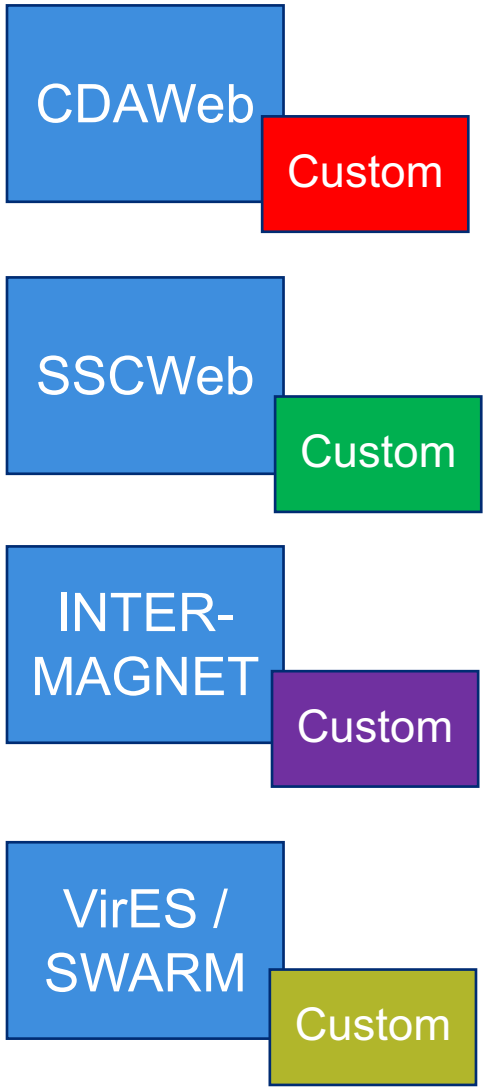
Retrieve metadata for dataset

Stream data

HAPI Offers many benefits

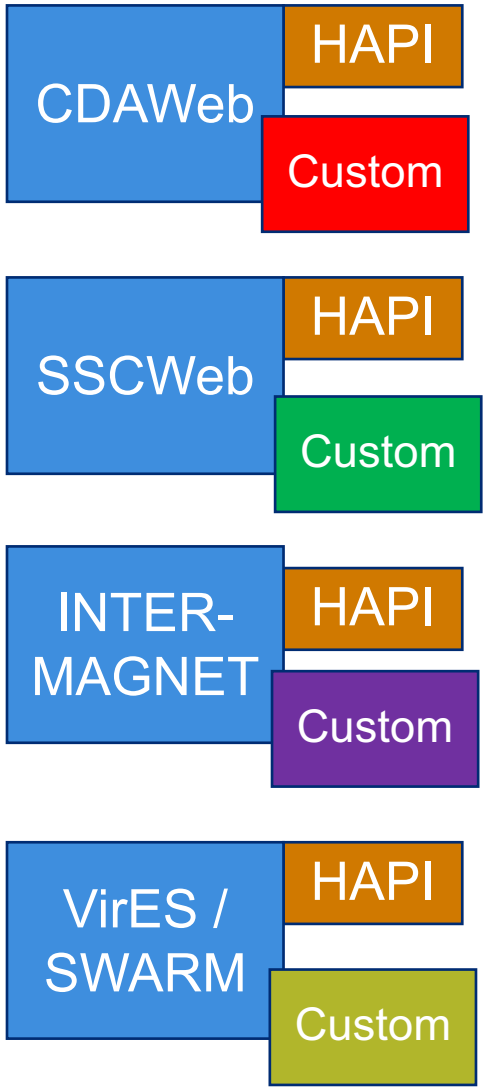
For Scientists / Data Users

- having a standard access mechanism means less data wrangling code
  - HAPI client library functions like a HDF library or FITS reading library (gets you to the data)
  - HAPI data sources have a known level of documentation
  - HAPI is becoming more common at multiple places
- solves the “fill my array” problem
- can share HAPI URLs as a way to communicate data
- can share tools more easily:
  - data preparation tools since they all pull from



Many data providers have their own computer-based access mechanism (i.e., it's own API).

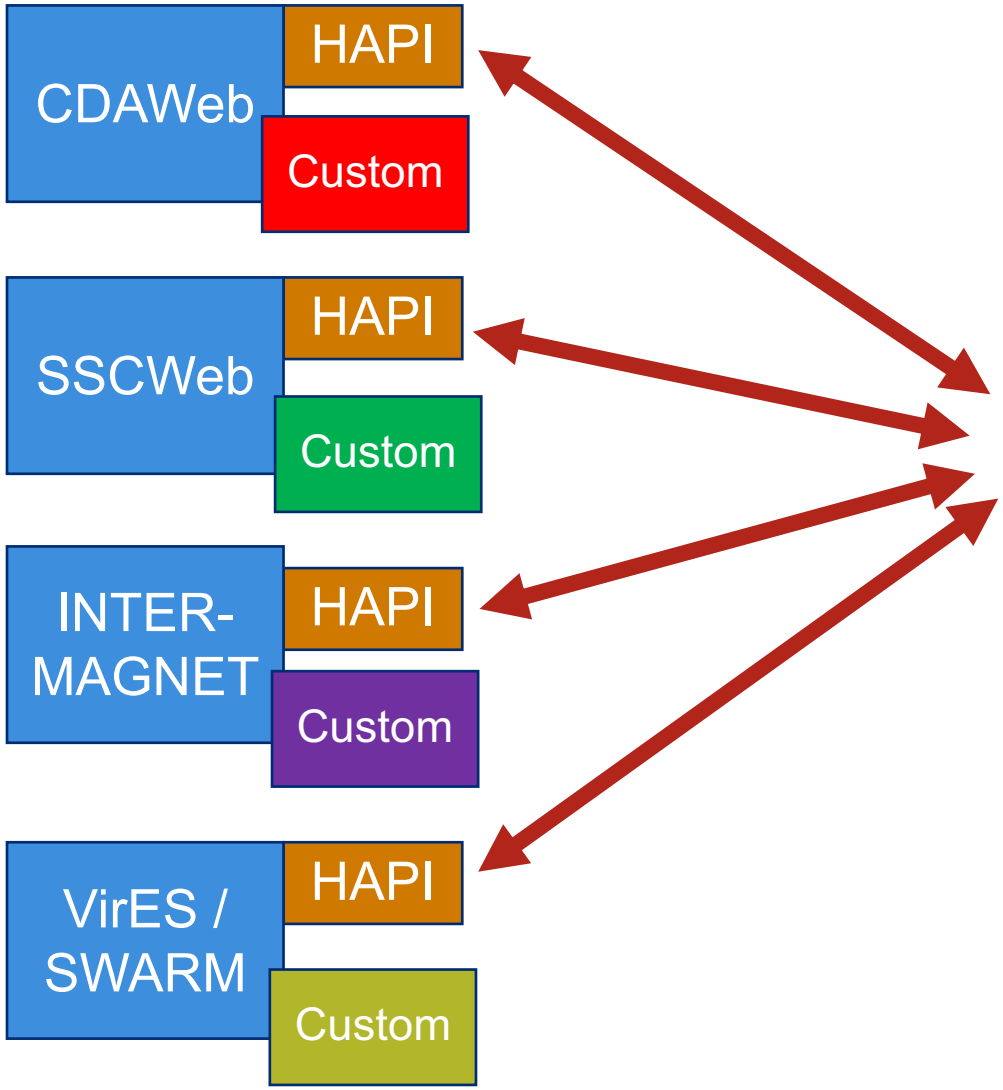
But these services have very similar in core features, and often only differ in almost trivial ways.



Many services hacve ADDED another way to access data using HAPI.

Note too that most services do not use our HAPI server code – they simply tweak their own server code to make it HAPI compliant.

This is evidence that HAPI merely represents a natural standardization of what people were already doing.



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

We do offer HAPI clients in many languages, and these should be re-used by projects since they hide the HAPI details from users and simply perform the “fill my array” role.

HAPI Offers Many Benefits

For Scientists / Data Users

- having a standard access mechanism means less data wrangling code
  - HAPI client library functions like a HDF library or FITS reading library (gets you to the data)
  - HAPI data sources have a known level of documentation
  - HAPI is becoming more common at multiple places
- solves the “fill my array” problem
- can share HAPI URLs as a way to communicate data
- can share tools more easily:
  - data preparation tools can possibly be made common since they all pull from the same API

HAPI Offers Many Benefits

For Data Providers:

- offering a standard access mechanism means that a basic aspect of your data delivery design (the API) is done for you
- If you use HAPI:
  - there are some data server codes you can use or adapt (several languages: Java, Python, JavaScript)
  - **there is a verifier capability to determine if you implemented the HAPI spec correctly**
  - **your users have clients they can start using right away**
  - these clients can also be used by your for testing or analysis
  - your users can more easily combine your data with other HAPI Data

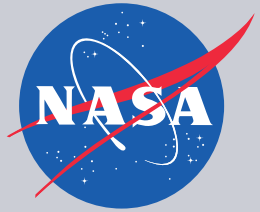
What are the long-term prospects of HAPI?

- the specification is simple and language-independent
  - i.e., not too hard to re-implement servers or clients in a new language once Python ages out
- all aspects are open source (the spec, the servers, the clients)
- HAPI Python library is a PyHC core package
- HAPI is a COSPAR standard
- other projects are also now developing clients, some of which may be useful to your and your users
- multiple other data sources are accessible the same way
  - your data can be more easily combined with other data sources

- multiple other space weather projects now looking at how to adopt HAPI:
  - future adopters: MADRIGAL, SuperMAG, AMPERE, HamSCI, Space Environment Canada
- HAPI is also applicable for any time series data
  - model output
  - spacecraft housekeeping
  - earthquake data
  - streamflow versus time
  - gridded global parameters

HAPI offers a way to achieve lasting interoperability for Heliophysics data analysis.

What data is available now with HAPI?

Institution	Server	Type of Data	Num of Datasets
	CDAWeb	Heliophysics	2800
	SSCWeb	Ephemeris	250
	SDAC	Solar Images (URLs)	50??
	CCMC	Space Weather Indices	250
IRAP Plasma Data Ctr	AMDA	Helio. & 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
Royal Netherlands Meteorological Institute	KNMI	Space Weather	~100
	ESAC / Cluster Mission Data	Helio. (magnetosphere)	

Coming soon

Institution	Server	Type of Data	Number of Datasets
esa	ESAC Solar Orbiter and others	Heliophysics	lots
JHU / APL	SuperMAG	global ground mag	~500
JHU / APL	TIMED / GUVI	ionospheric images	~10
NASA	PDS PPI Node	Planetary Plasma, Particle, and Fields	~1000

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

HAPI Clients – all open source

Libraries

- Python library (Bob Weigel)
- IDL library (Scott Boardsen)
- Java library (Jeremy Faden, Larry Brown)
- Matlab 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)
- KNMI visualizer (JavaScript, Eelco Doornbos)

Uses HAPI, but not fully open source: LASP Space Weather Data Portal (JavaScript, Jenny Knuth)

Not just line plots: higher dimensional data is supported by HAPI

2014-06-08

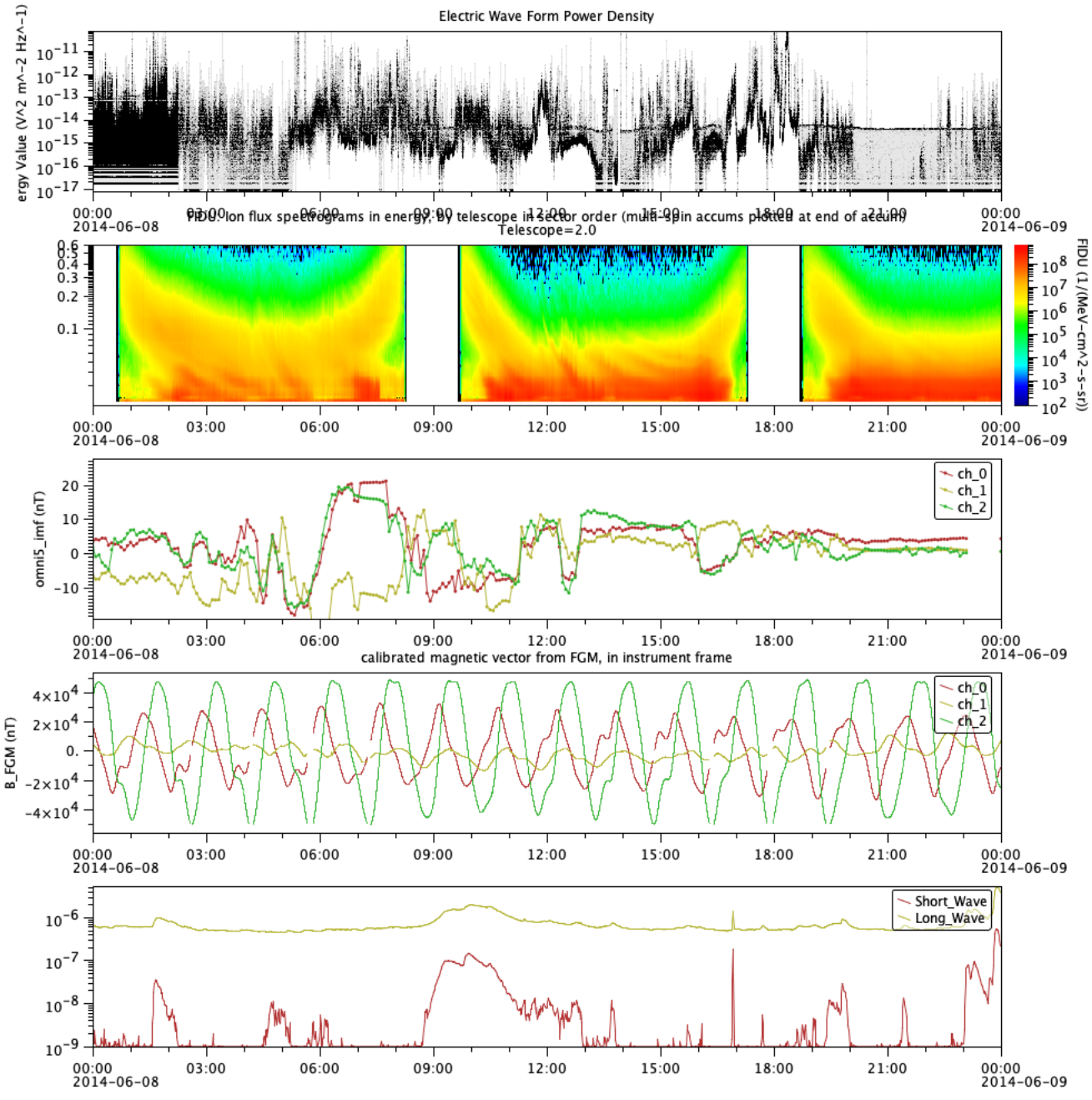
ESAC HAPI Server
Cluster-3 Electric Wave Form Power Density

CDAWeb HAPI Server
Van Allen Probes, RBSPICE Ion Energy Spectrogram

AMDA HAPI Server
Omni MAG data

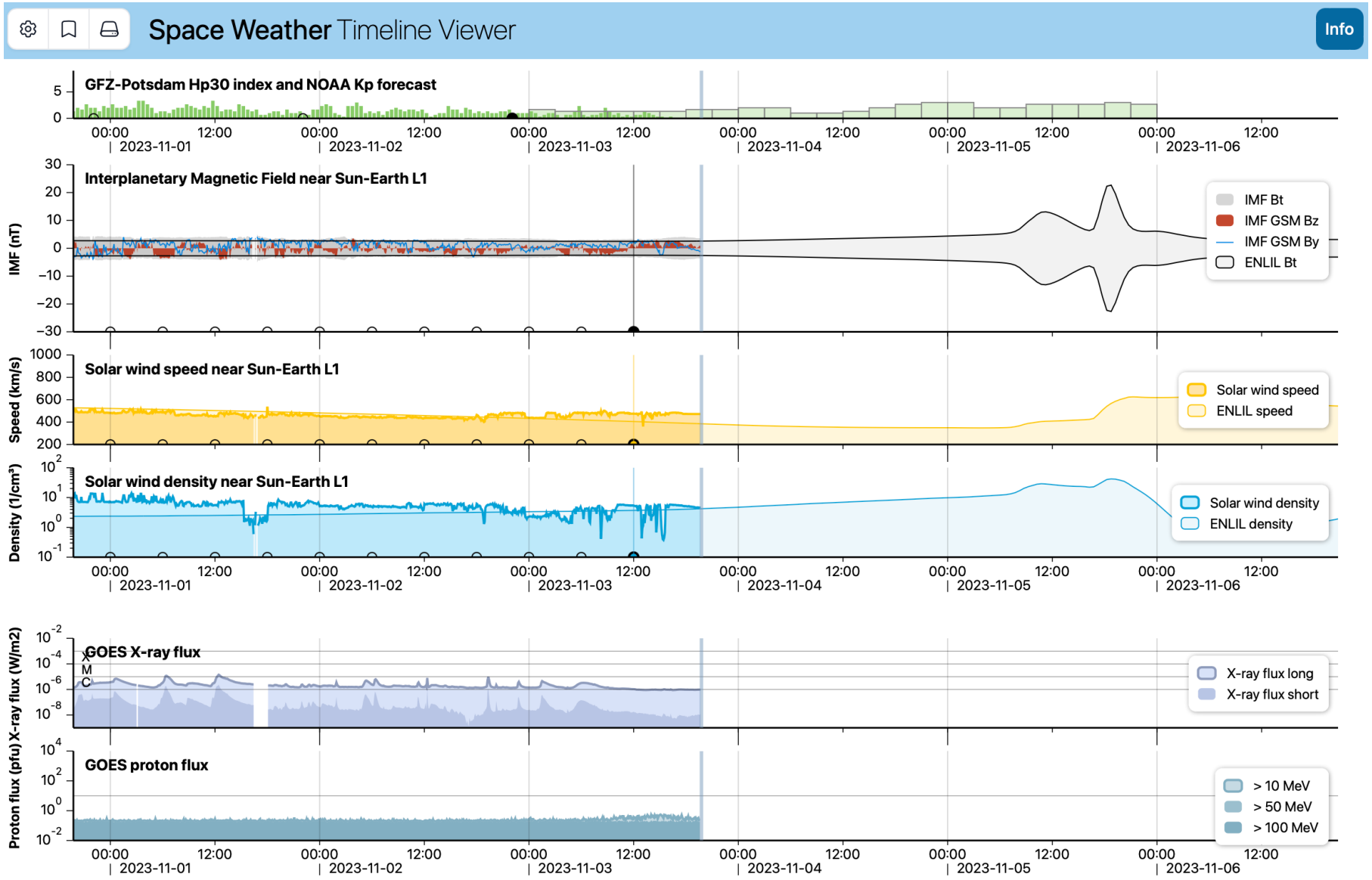
ViRES HAPI Server
GRACE-A MAG Data

CCMC ISWAT HAPI Server
GOES X-ray flux



Royal Netherlands Meteorological Institute (KNMI) Space Weather Timeline Viewer

HAPI server and open-source Javascript HAPI client both developed externally to the HAPI team!



HAPI Servers – all open source

- Most mature: a node.js server that handles all HAPI front end communication and has a flexible command-line plugin structure for retrieving local data
- Python server – has been used, but not in production right now
- Java server – being revamped for CDAWeb use and also at ESAC

(recall – most server implementors rolled their own, since the HAPI protocol was not too different form their existing API)

HAPI Server Verifier

- For server developers, there is a mechanism to test a new or existing server
- The test can re run online (using our server that runs the test code) or locally (download the test code and run it yourself)
- It tests your server to look for compliance with the specification

Demonstrations

- Java-script client in a web page
  - http://hapi-server.org/servers
- Display client (written outside the team by Eelco Doornbos at KNMI)
  - https://spaceweather.knmi.nl/viewer/?layout=viirs
- (maybe in Q&A) Python client for obtaining data for custom analysis (Bob Weigel)
  - https://colab.research.google.com/github/hapi-server/client-python-notebooks/blob/master/hapi_demo.ipynb#examples
  - https://github.com/hapi-server/client-python
- Other options to learn:
  - Tutorial notebooks from PyHC Summer School
  - https://github.com/hapi-server/tutorial-python
  - Start at 1:36:00 in this video for a guide through some HAPI exercises / tutorials:
  - https://www.youtube.com/watch?v=G_UuTP7OMZg&list=PLDKhoNyHGTFZ345-ll-EeC4CAQhNUfUS0&index=3

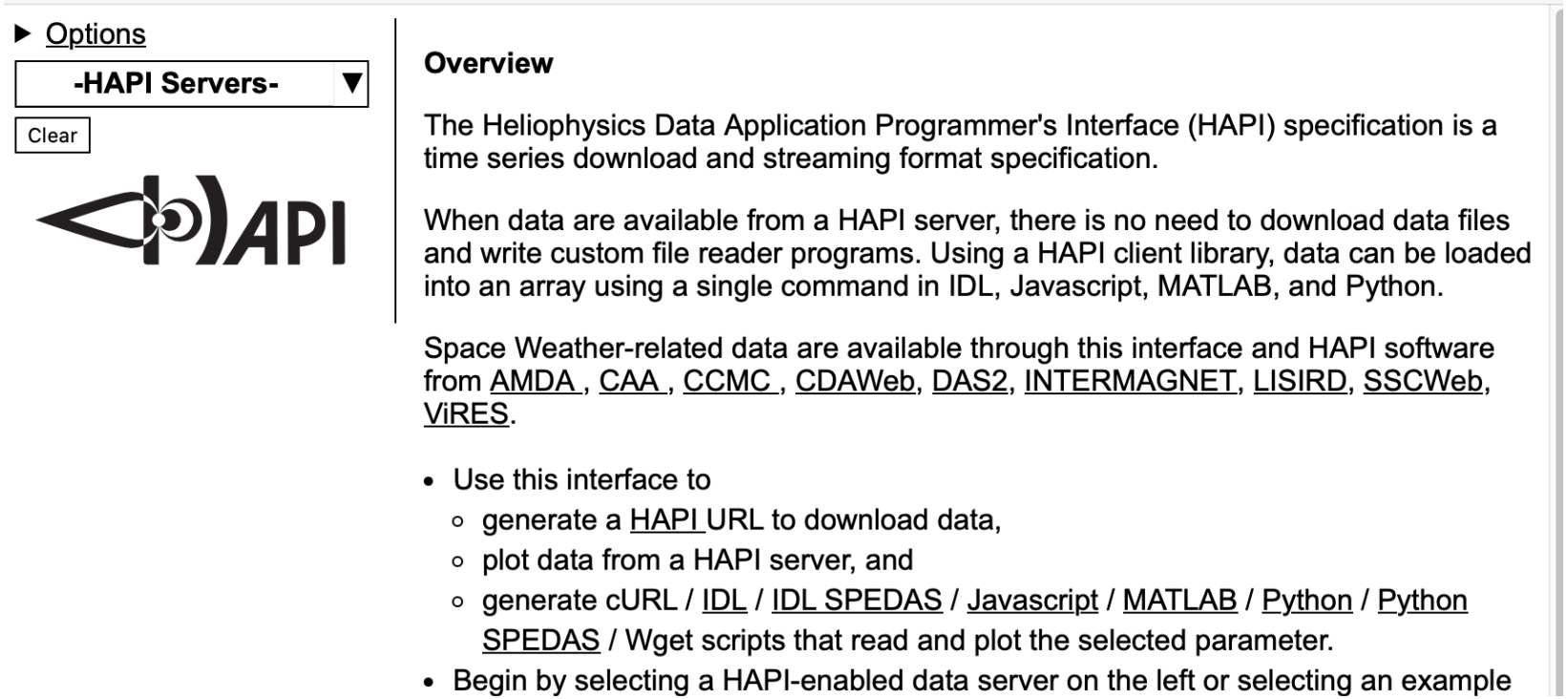
HAPI Access Illustrated using a JavaScript client

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

Explore how a simple HAPI client interacts with HAPI servers.

This heavily human-interaction approach is not the intended primary way to use HAPI, but it allows you to see what data is available.

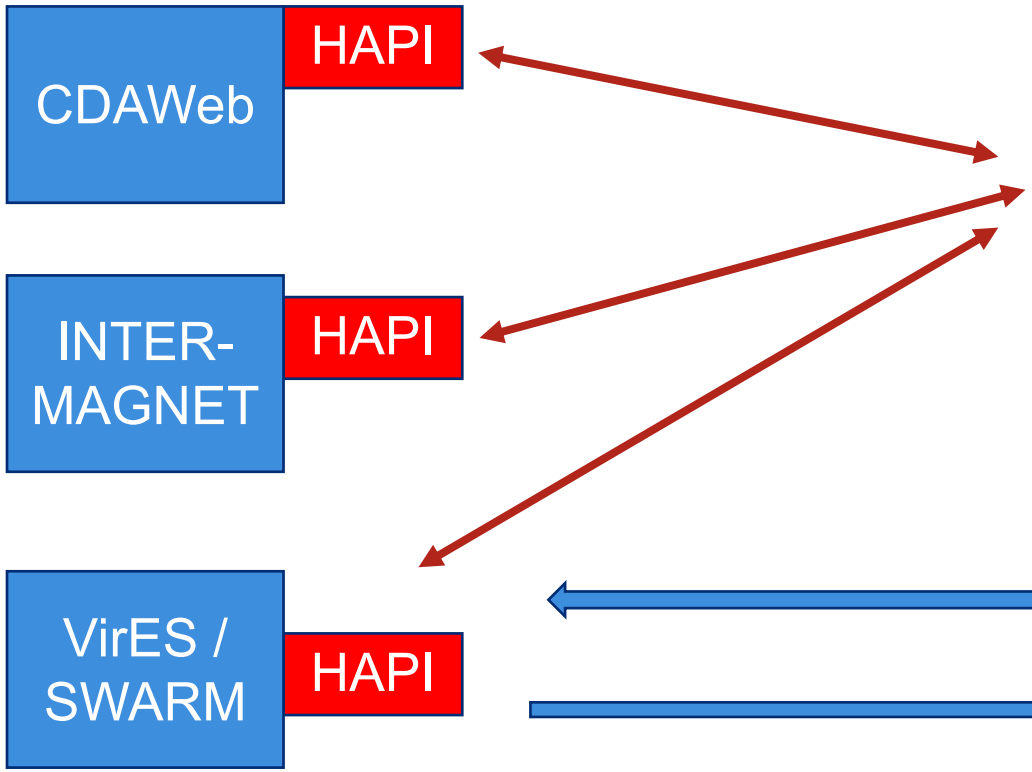
It can create code stubs for you to get started (Python, IDL, Matlab, etc)



Illustrating HAPI Access with a web-based client

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!

Python code stub generated for you

```
# 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)
```

Reference page – plenty of resources

- http://hapi-server.org
  - one-page summary, video intro, links to Github repos, mailing lists, etc
- R. Weigel et al, HAPI: An API Standard for Accessing Heliophysics Time Series Data
  - JGR Space Physics: https://doi.org/10.1029/2021JA029534
- **The HAPI Specification**
  - Version 3.0 https://doi.org/10.5281/zenodo.4757597
  - The development location for the latest spec and drafts and open issues, etc: https://github.com/hapi-server/data-specification
- Lots of other open source HAPI projects:
  - https://github.com/hapi-server
- **Try it out with this interactive way to explore known HAPI servers!**
  - http://hapi-server.org/servers/
- Tutorial Notebooks for using HAPI in Python:
  - https://github.com/heliophysicsPy/summer-school/tree/main/hapi-tutorial

What if every time series dataset was accessible through a single API?

Read the Specification

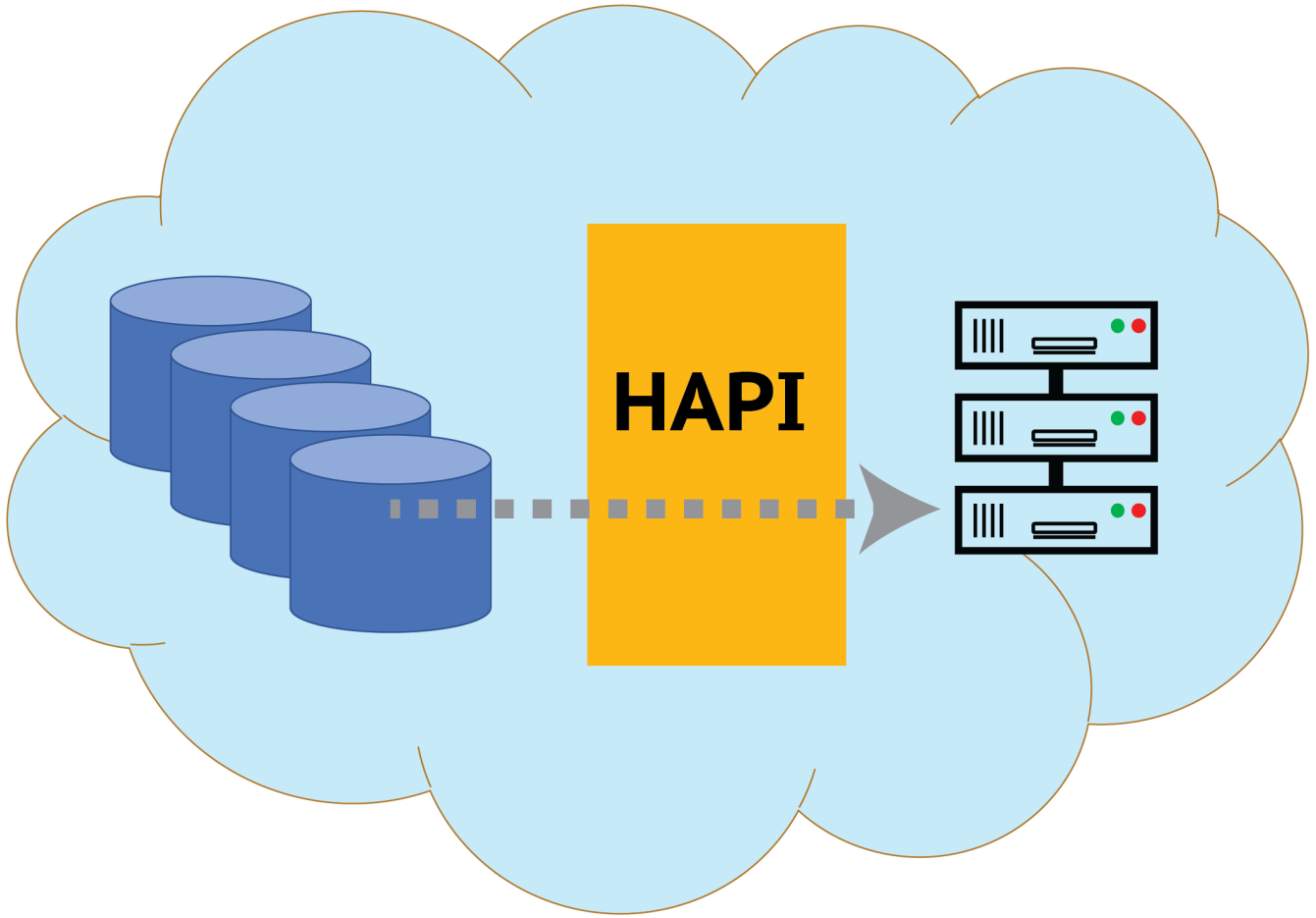
Try getting data! Live Link to a HAPI Data Request (for mag data at CDAWeb)

```
https://cdaweb.gsfc.nasa.gov/hapi/data?
   id=AC_K2_MFI&time.min=2023-01-01&time.max=2023-01-02
```

Backup and More Details

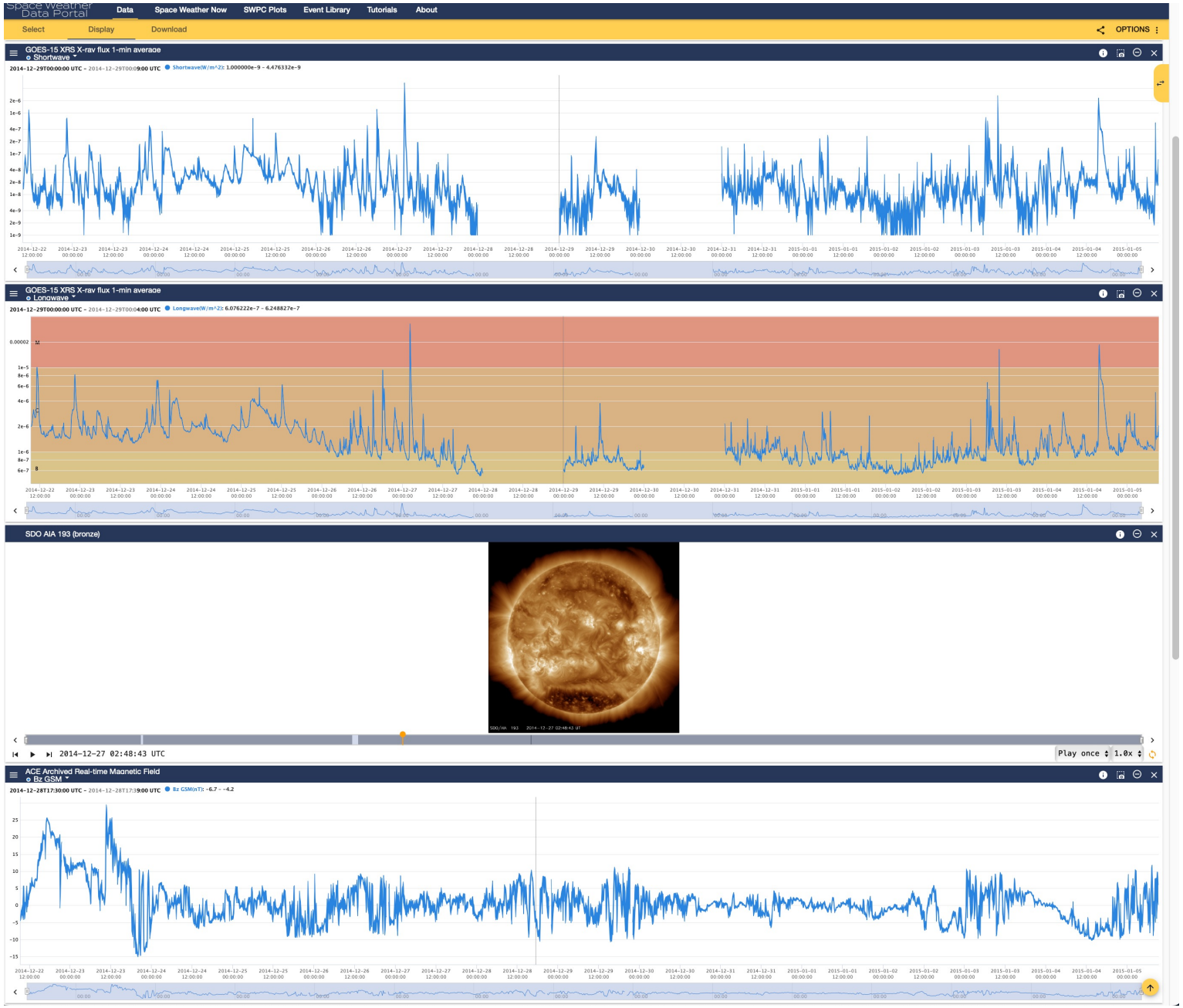
HAPI in the Cloud

HAPI also works serving data from a cloud store to cloud computing resources (no egress). It can also egress data to non-cloud environments.

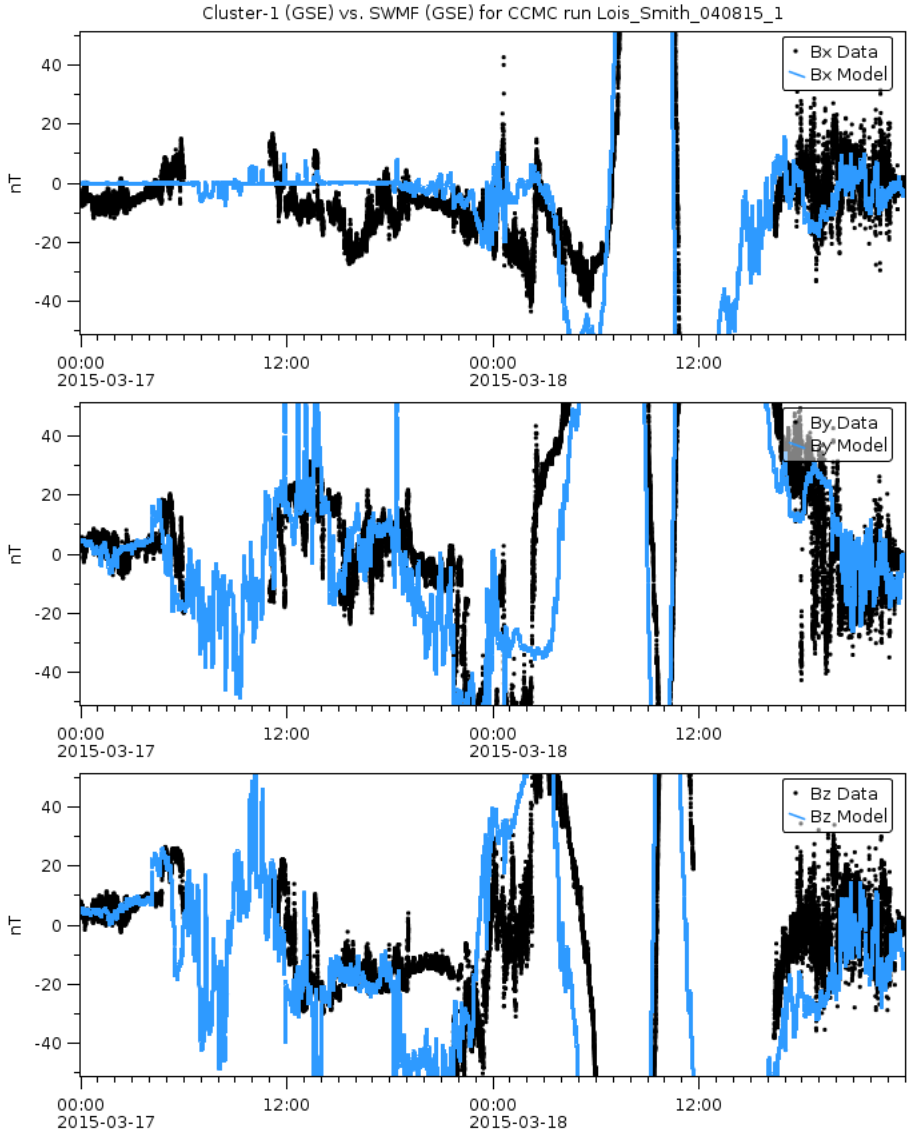


LASP Space Weather Viewer

Was trivial to add HAPI capability so that any HAPI source can also be displayed.



CCMC example: Data-Model Comparisons



Steps:

- Use model data from SWMF.
- Get Cluster-1 **ephemeris from SSCWeb HAPI server**.
- Fly the Cluster-1 trajectory through the model output to get mag data along this trajectory
- Get Cluster-1 mag **data from CDAWeb HAPI server.**

This example is from the CCMC web page:
https://ccmc.gsfc.nasa.gov/ungrouped/GM_IM/GM_analysis.php?Pid=12870&Pt=BO&Ps=Cluster-1
Work is by Darren De Zeeuw.

HAPI is a COSPAR-endorsed standard for Space Weather data access

COSPAR Panel on Space Weather Resolution on Data Access

Accepted at COSPAR PSW Business Meeting on 18 July 2018 (updated 15 October 2021).

Taking into account that:
1. It is in the general interest of the international heliophysics and space weather community that data be made as widely accessible as possible,

2. The open exchange of data benefits from well-defined and standardized methods of access,

3. The ILWS-COSPAR Roadmap has recommended to standardize metadata and harmonize access to data and model archives, and

4. The Heliophysics Application Programmer's Interface (HAPI) specification has demonstrated that it is comprehensive and can meet the needs of the community,

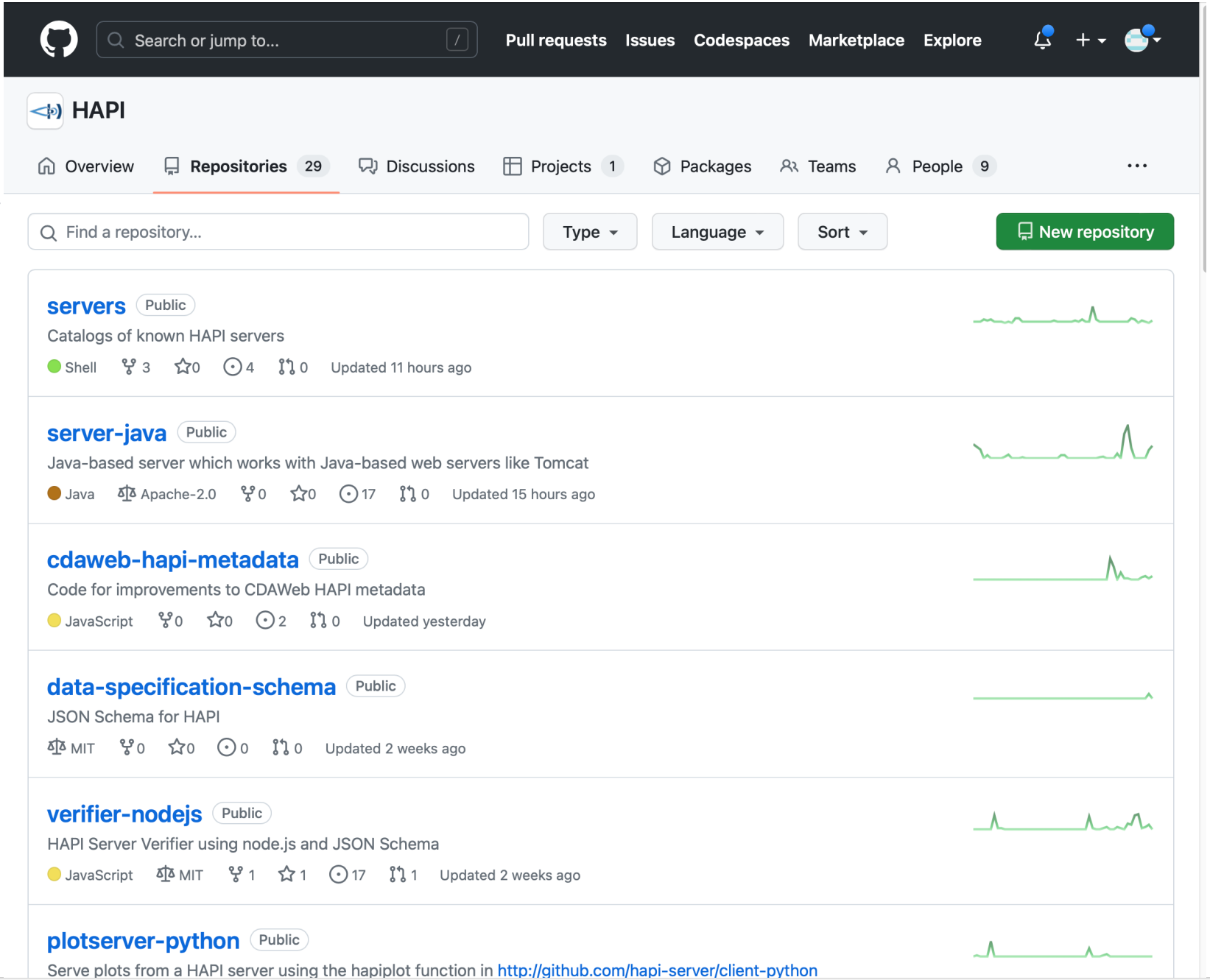
The COSPAR PSW resolves that there is a need for at least one common data access API to facilitate and enhance international access to data.

Therefore, it is recommended that:

1. HAPI https://doi.org/10.5281/zenodo.4757597) be the common data access API for space science and space weather data.
2. Funding agencies provide encouragement and adequate support to enable data produced by projects to be accessed by using HAPI compliant services.

29 projects in GitHub

`https://github.com/hapi-server`



List of known HAPI servers

- https://github.com/hapi-server/servers/blob/master/all.txt

```
http://hapi-server.org/servers/SSCWeb/hapi
https://cdaweb.gsfc.nasa.gov/hapi
https://imag-data.bgs.ac.uk/GIN_V1/hapi
https://iswa.gsfc.nasa.gov/IswaSystemWebApp/hapi
http://lasp.colorado.edu/lisird/hapi
http://hapi-server.org/servers/TestData2.0/hapi
https://amda.irap.omp.eu/service/hapi
https://vires.services/hapi
https://api.helioviewer.org/hapi/Helioviewer/hapi
```

Some development servers (either for testing or nearly ready for science):

https://github.com/hapi-server/servers/blob/master/dev.txt

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.”

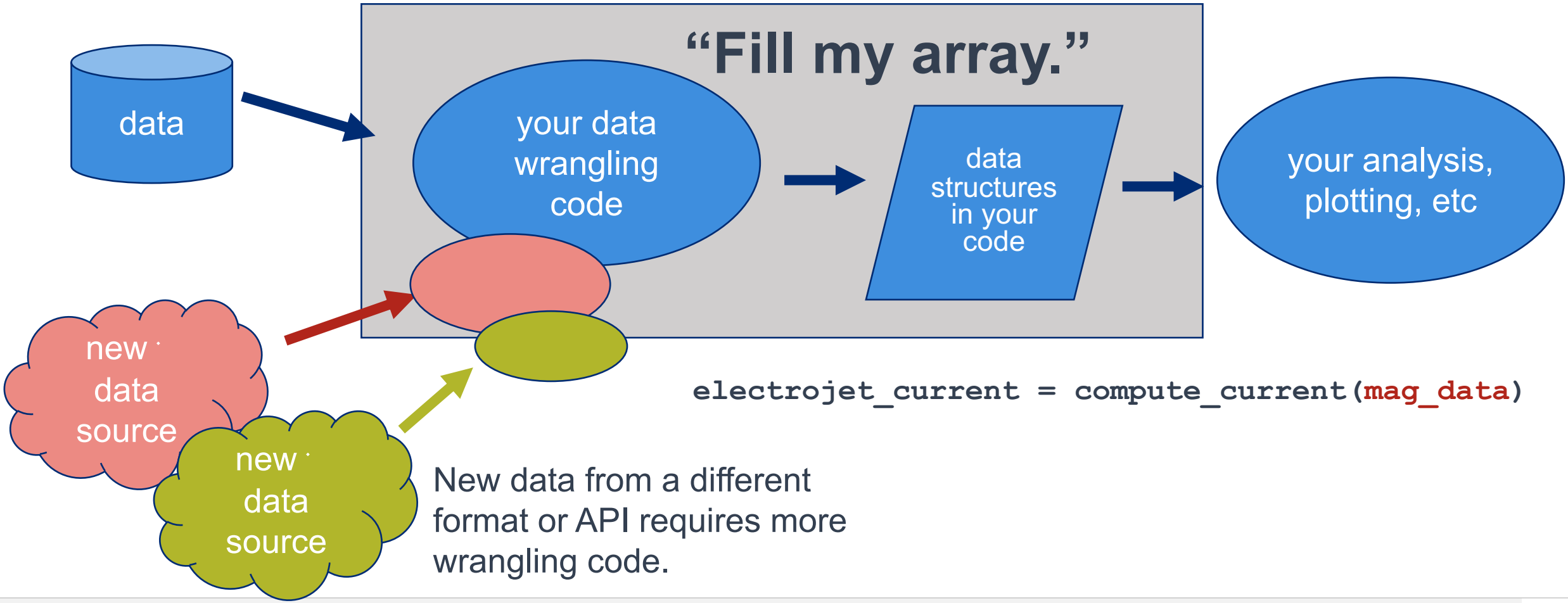
```
# some kind of read routine to obtain data:
mag_data = read_data( dataset, selection_criteria )

# 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.”



HAPI – a very simple request and response

Request:

```
http://server.org/hapi/info?dataset=ACE_MAG
```

(this assumes you already got dataset IDs from the catalog endpoint)

HAPI – a very simple request and response

Request:

```
http://server.org/hapi/info?dataset=ACE_MAG
```

(this assumes you already got dataset IDs from the catalog endpoint)

Response:

```
{
   block of JSON metadata, including:

   time range covered by data

   lists of parameters that can be requested

   can also include other metadata
        either by reference or inclusion
}
```

HAPI `info/` response example (JSON metadata)


```
{
  "HAPI": "2.0",
  "status": {"code": 1200, "message": "OK"},
  ”startDate": "1997-09-02T00:00:00Z",
  "stopDate": "2023-08-20T23:00:00Z",
  "resourceURL": "https://cdaweb.gsfc.nasa.gov/misc/NotesA.html#AC_H2_MFI",
 "contact": "N. Ness @ Bartol Research Institute”   "parameters": [
    { "name": "Time", "type": "isotime", "units": "UTC", "length":24, "fill": null },
    { "name": "Magnitude", "type": "double", "units": "nT", "fill": "-1.0E31",
            "description": "B-field magnitude" },
   { "name": "BGSEc", "type": "double", "units": "nT", "fill": "-1.0E31", "size": [3],
            "description": "Magnetic Field Vector in GSE Cartesian coordinates (1 hr)"},
   { "name": "BGSM", "type": "double", "units": "nT", "fill": "-1.0E31", "size": [3],
            "description": "Magnetic field vector in GSM coordinates (1 hr)", },
   { "name": "SC_pos_GSE", "type": "double", "units": "km", "fill": "-1.0E31", "size": [3],
            "description": "ACE s/c position, 3 comp. in GSE coord."},
   { "name": "SC_pos_GSM", "type": "double", "units": "km", "fill": "-1.0E31", , "size": [3],
            "description": "ACE s/c position, 3 comp. in GSM coord.” } ],
}
```

Try HAPI access using these QR codes

- Info response:
  https://cdaweb.gsfc.nasa.gov/hapi/info?id=AC_K2_MFI

- Data Response:
  https://cdaweb.gsfc.nasa.gov/hapi/data?id=AC_K2_MFI&time.min=2023-01-01&time.max=2023-01-02

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)

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