JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

The HAPI Protocol for Time Series Access

(Heliophysics Application Programmers Interface)

Jon Vandegriff, Johns Hopkins Applied Physics Lab
Robert Weigel, George Mason University
Todd King, University of California Los Angeles
Jeremy Faden, Cottage Systems / University of Iowa
Aaron Roberts, NASA Goddard Space Flight Center
Bernard Harris, NASA Goddard Space Flight Center
Robert Candey, NASA Goddard Space Flight Center
Nand Lal, NASA Goddard Space Flight Center
Scott Boardsen, NASA Goddard Space Flight Center
Doug Lindholm, Colorado University / LASP
Larry Brown, Johns Hopkins Applied Physics Lab
Eric Grimes, University of California Los Angeles

get a QR code reader ready
on your phone for a demo

What problem does HAPI address?

- focus is tabular, high density time series data
- meant for data access, not discovery
  - light on metadata, with optional links to more
  - can add more metadata keyword/value pairs
- exposes data content of time series data sets
- allows query for arbitrary* time ranges of data, regardless of storage format

Terminology check: "dataset"

Heliophysics usage:

dataset: a mission-length collection of data (often in daily or otherwise periodic files) containing a uniform set of records, where each record contains a set of parameters

parameter: measured quantity or related ancillary value; one data element in a record; also referred to as "variables"

Terminology check: "time series data"

- conceptually a table, like a spreadsheet, but potentially millions of points (sub-second resolution over years)
- time column with any number of **parameters**, with one parameter per data column
- each parameter can be multidimensional
- multi-dimensional parameters that are spectral in nature need to have the spectral ranges defined

Time	data1	scalar2	array	multiDimArray
t0	d0	s0	a0[11]	m0[3,8]
t1	d1	s1	a1[11]	m1[3,8]
t1	d2	s2	a2[11]	m2[3,8]
t2	d3	s3	a3[11]	m3[3,8]
t4	d4	s4	a4[11]	m4[3,8]
t5	d5	s5	a5[11]	m5[3,8]
t6	d6	s6	a6[11]	m6[3,8]
…	…	…	…	…

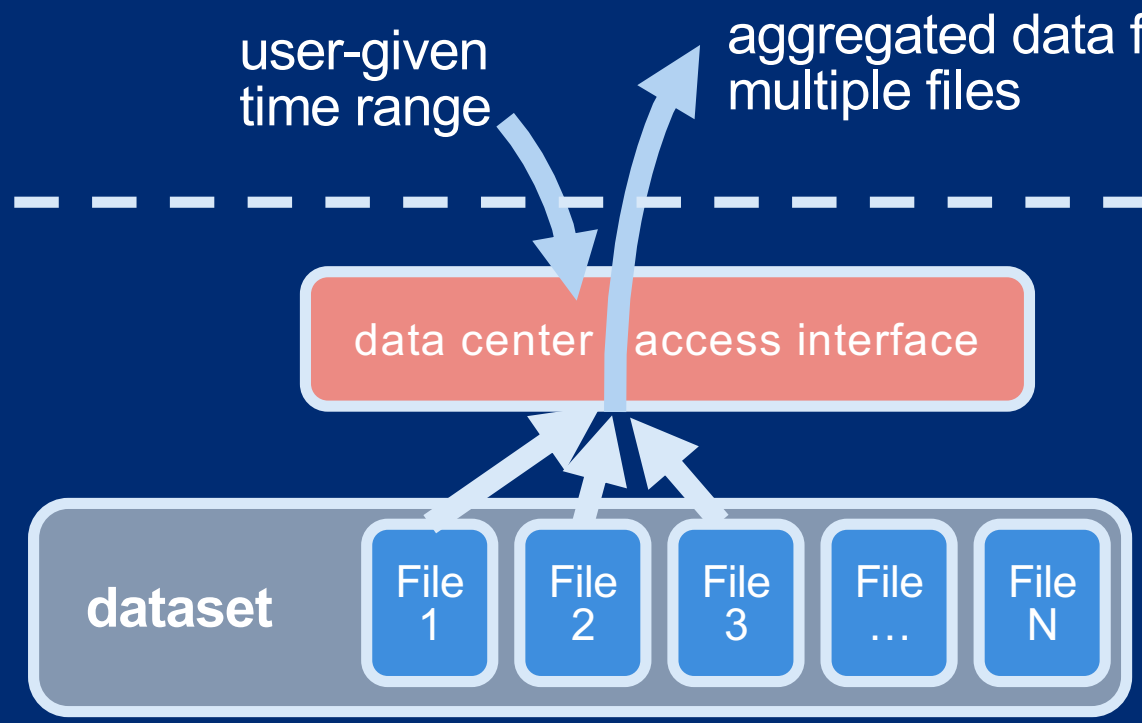
*think of the entire **dataset** as one giant table*

metadata for the "a" parameter might need to describe the spectral ranges of the 11 "bins";

and similarly for the 3x8 bins of the "m" parameter

HAPI: a uniform interface for Timeseries Data Delivery

Many data centers offer a service to request data content programmatically (computer-to-computer)

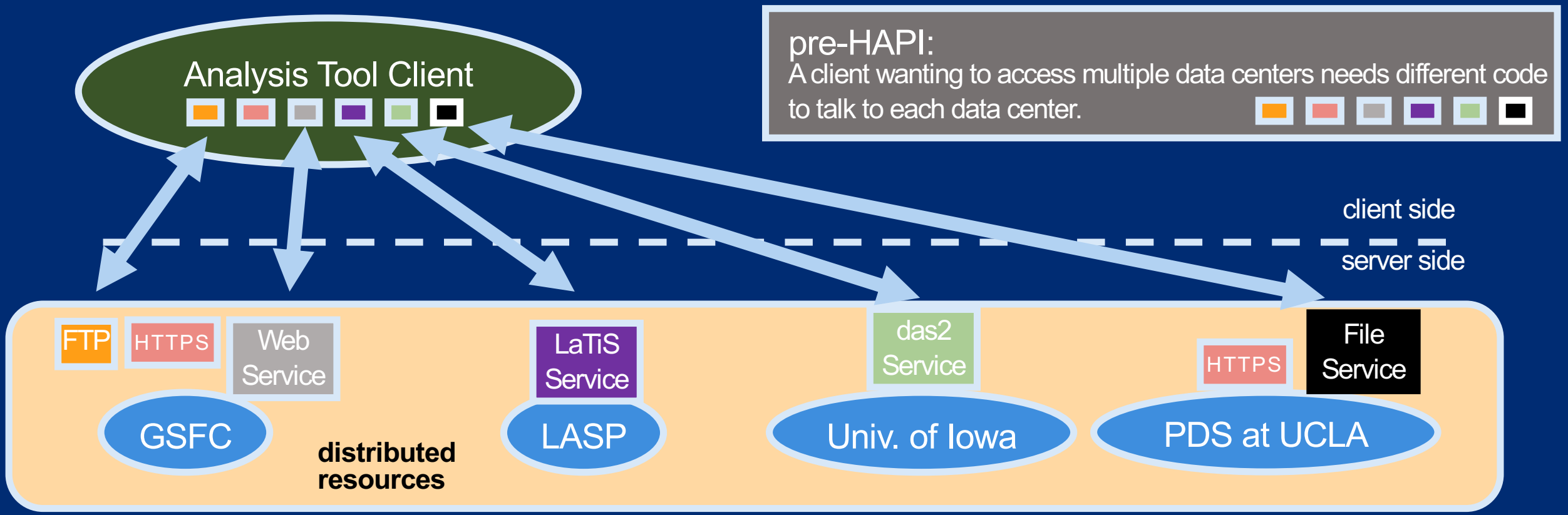


the aggregate can be:
a single merged file, or
a stream;

for HAPI, it is a stream

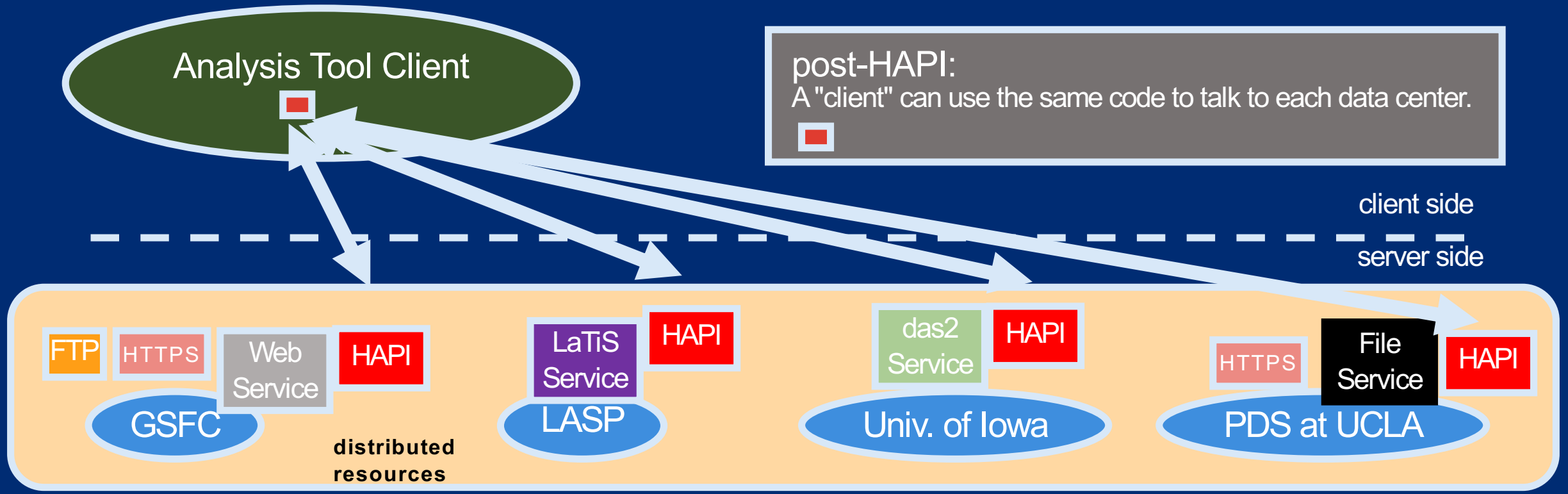
Current data systems are similar but not identical

- many Heliophysics data centers have an access API
- they are similar, but not API compatible



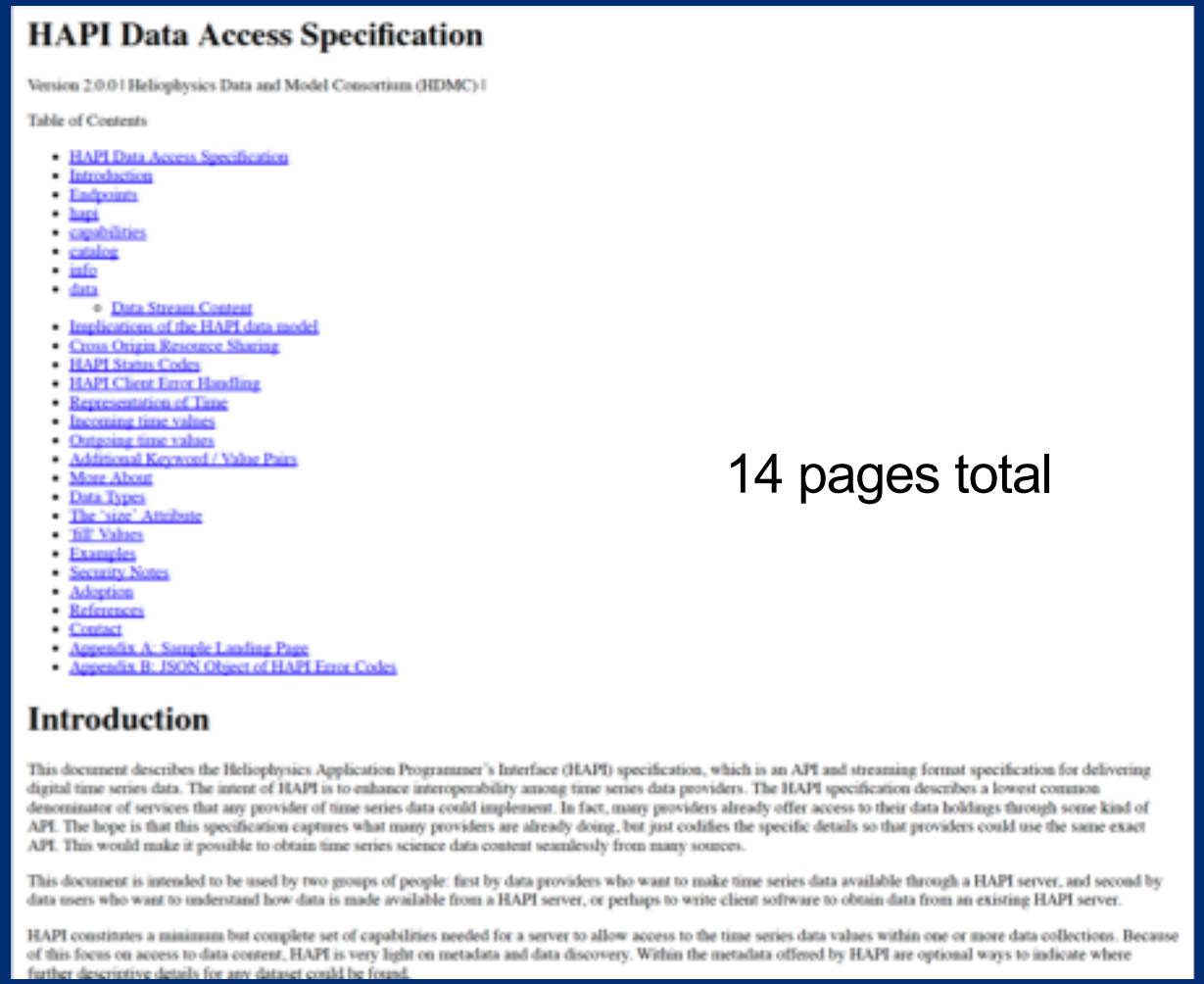
HAPI captures the lowest common denominator

- focus of HAPI is plain access with nothing fancy (no filtering, averaging, etc)
- often, HAPI can be integrated into an existing service's code base



HAPI (Heliophysics Application Programmer's Interface)

Main product is a specification



HAPI Data Access Specification

Version 2.0.0 | Heliophysics Data and Model Consortium (HDMC) |

Table of Contents

Introduction

This document describes the Heliophysics Application Programmer's Interface (HAPI) specification, which is an API and streaming format specification for delivering digital time series data. The intent of HAPI is to enhance interoperability among time series data providers. The HAPI specification describes a lowest common denominator of services that any provider of time series data could implement. In fact, many providers already offer access to their data holdings through some kind of API. The hope is that this specification captures what many providers are already doing, but just codifies the specific details so that providers could use the same exact API. This would make it possible to obtain time series science data content seamlessly from many sources.

This document is intended to be used by two groups of people: first by data providers who want to make time series data available through a HAPI server, and second by data users who want to understand how data is made available from a HAPI server, or perhaps to write client software to obtain data from an existing HAPI server.

HAPI constitutes a minimum but complete set of capabilities needed for a server to allow access to the time series data values within one or more data collections. Because of this focus on access to data content, HAPI is very light on metadata and data discovery. Within the metadata offered by HAPI are optional ways to indicate where further descriptive details for any dataset could be found.

14 pages total

All HAPI-related projects are online at:

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

Key Elements of the HAPI Spec

- request interface – how do clients request data (and metadata)

  4 REST-ful endpoints, each with simple parameters (or none)

  capabilities
  catalog
  info (1 parameter, the dataset name)
  data (multiple parameters: time range, dataset name, desired parameters)

- response interface – what is the format of the returned data stream (and metatdata)
  a. uniform view of the data, regardless of it storage format on the server
  b. time column is always the first column
  c. time format always the same
  d. simple data layout that also allows for multi-dimensional data
  e. contains spectral bin parameters (energy ranges, frequency ranges, etc)

HAPI is very light on metadata;
The focus is data delivery with the minimum metadata to make the data scientifically useful

HAPI Request / Response Example

http://datashop.elasticbeanstalk.com/hapi/**catalog**

```
{ "HAPI": "2.0",
  "status": { "code": 1200, "message": "OK" }

  "catalog": [
    { "id": "CAPS_ELS_FLUX_HIRES" },
    { "id": "CASSINI_LEMMS_PHA_CHANNEL_1_MIN_AVG" },
    { "id": "CASSINI_LEMMS_PHA_PITCH_ANGLES_10_MIN" },
    { "id": "CASSINI_MAG_HI_RES" },
    { "id": "CHEMS_PHA_BOX_FLUXES_FULL_TIME_RES" },
    { "id": "CHEMS_RATES_FLUXES_FULL_TIME_RES" },
    { "id": "LEMMS_PHA_CHANNELS_FULL_TIME_RES" },
    { "id": "LEMMS_PRIORITY_CHANNELS_FULL_TIME_RES" },
    { "id": "LEMMS_REGULAR_CHANNELS_FULL_TIME_RES" }
  ],
}
```

catalog

HAPI Request / Response Example

info

http://datashop.elasticbeanstalk.com/hapi/**info?id=CASSINI_MAG_HI_RES**

(this is not complete!)


```
{ "HAPI": "2.0",
  "status": { "code": 1200, "message": "OK" },
  "stopDate": "2017-258T10:31:10.425Z",
  "startDate": "2004-001T00:00:04.734Z",
  "sampleStartDate": "2004-183T00:00:00.000Z",
  "sampleStopDate": "2004-184T00:00:00.000Z",
  "description": "Cassini magnetometer data as used by the MIMI team",
  "cadence": "PT5S"
  "parameters": [
    {"name": "epoch", "type": "isotime", "length": 24, "units": "UTC", "fill": null,
      "description": "time as ISO 8601 UTC string to milliseconds"  },
    {"name": "Dist_Rs", "type": "double", "units": "Saturn_radii", "fill": null,
      "description": "Cassini to Saturn distance" },
    {"name": "Bx_SSO", "type": "double", "units": "nt", "fill": "-1.0e+38",
      "description": "x component of magnetic field in the SSO frame" },
    {"name": "By_SSO", "type": "double", "units": "nt", "fill": "-1.0e+38",
      "description": "y component of magnetic field in the SSO frame" },
    {"name": "Bz_SSO", "type": "double", "units": "nt", "fill": "-1.0e+38",
      "description": "z component of magnetic field in the SSO frame" }
  ]
}
```

HAPI Request / Response Example

http://datashop.elasticbeanstalk.com/hapi/**data?id=CASSINI_MAG_HI_RES&time.min=2004-183T00:00:00&time.max=2004-184T00:00:00&parameters=Bx_SSO,By_SSO,Bz_SSO&include=header**

data

(this is not complete!)

```
#{
#  "HAPI": "2.0",
#  "status": {"code": 1200,"message": "OK" },
#  "parameters": [
#    { "name": "epoch", "type": "isotime", "length": 24,
#      "units": "UTC", "fill": null,
#      "description": "time as ISO 8601 UTC string to milliseconds" },
#    { "name": "Bx_SSO", "type": "double", "units": "nt", "fill": "-1.0e+38" },
#    { "name": "By_SSO", "type": "double", "units": "nt", "fill": "-1.0e+38" },
#    { "name": "Bz_SSO", "type": "double", "units": "nt", "fill": "-1.0e+38" },
#  ],
#  "startDate": "2004-001T00:00:04.734Z",
#  "stopDate": "2017-258T10:31:10.425Z",
#  "description": "Cassini magnetometer data as used by the MIMI team",
#  "cadence": "PT5S",
#  "format": "csv"
#}
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
```

Existing Servers

Current list of operational servers is at HAPI Github site.

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

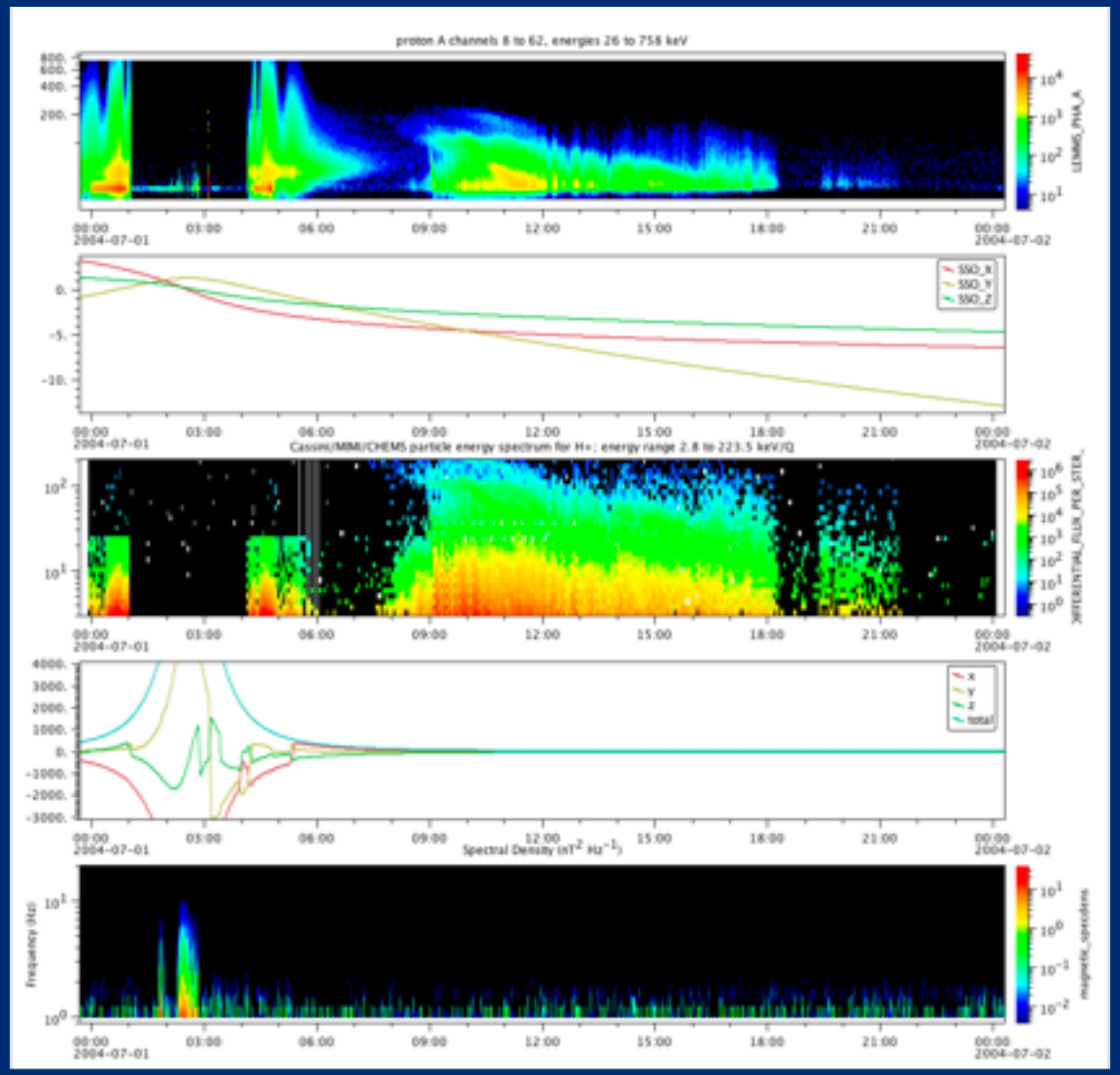
GSFC
University of Iowa
JHU/APL
CCMC
PDS
LASP (almost)

Existing Clients

Autoplot	`http://autoplot.org`
Python Client	`http://hapi-server.org/servers`
SPEDAS	`http://spedas.org`
stand-alone IDL client	`http://hapi-server.org/servers`
HAPI MIDL4	`http://cassini-mimi.jhuapl.edu/MIDL/HAPI/`

Autoplot with panels from several HAPI servers

2004-07-01 (day of year 183, Saturn orbit insertion)



energetic particles (LEMMS proton energy spectra)	from APL
Cassini trajectory data	from Iowa
energetic particles (CHEMS proton energy spectra)	from APL
Cassini MAG data	from APL
plasma wave data RPWS	from Iowa

Python Client for HAPI

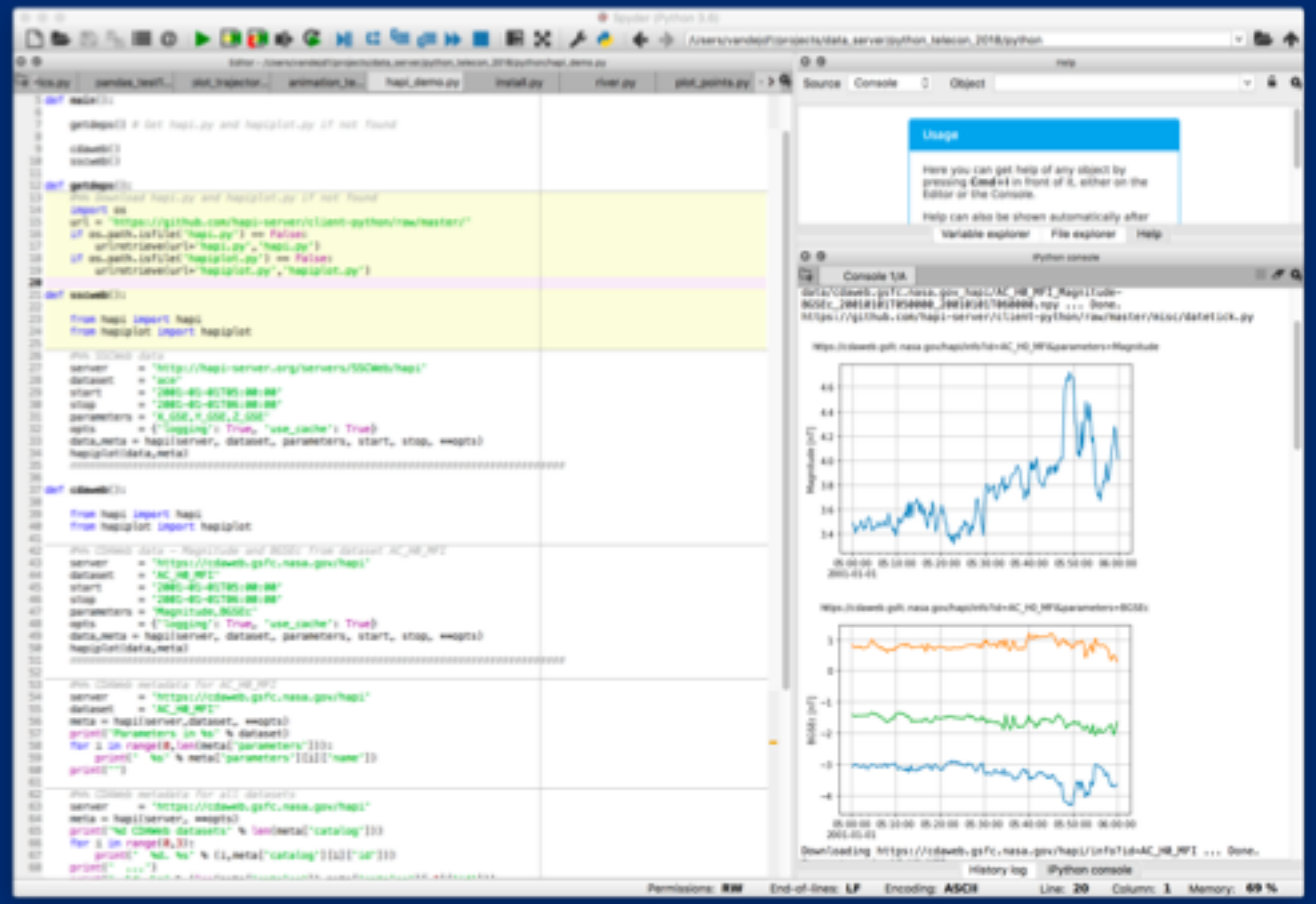
`https://hapi-server.github.io/` *top level HAPI page*

`https://github.com/hapi-server/client-python` *Python client page directly.*

Plots made of data from CDAWeb HAPI server by running a demo program.

Robert Weigel is coordinating the Python client effort.

Will be integrated into HelioPython (an emerging Python library).



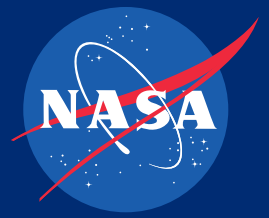
If you want to implement a HAPI server

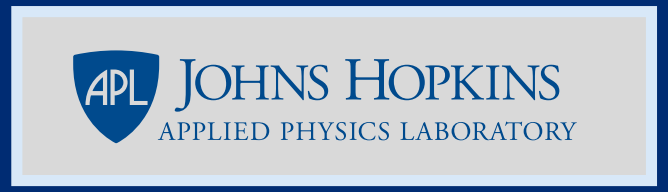
- read the specification -- it is very complete
- see if HAPI protocols can be added within an existing access mechanism at your site
- test your server
  - **use the HAPI verifier: http://tsds.org/verify-hapi**
    - this checks all of the important parts of the spec, and many corner cases
  - try your server in an existing client (Autoplot, SPEDAS, MIDL4)
- sign up on the mailing list: hapi-news@hapi-server.org
- ask for help
- email anyone involved in making the spec (authors on this paper)
- contact other early adopters

Next Steps

- **include others in discussions about specification and implementations**

- drop-in data server – still needs development (maybe LASP, maybe an Autoplot-based server, maybe something different)
- some additions to the spec for time-varying spectral bins
- integrate the Python client into emerging Heliophysics Python library
- other client development: IDL and Matlab enhancements
- encourage other data providers to use HAPI: NOAA (National Oceanographic and Atmospheric Administration)





Astronomy projects welcome…

hapi-news-bounces@hapi-server.org

JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

BACKUP SLIDES

Science Analysis with Distributed Data

or the Quest for Generic Interoperability

High level science workflow:

1. ask a question
construct and submit query to distributed resources

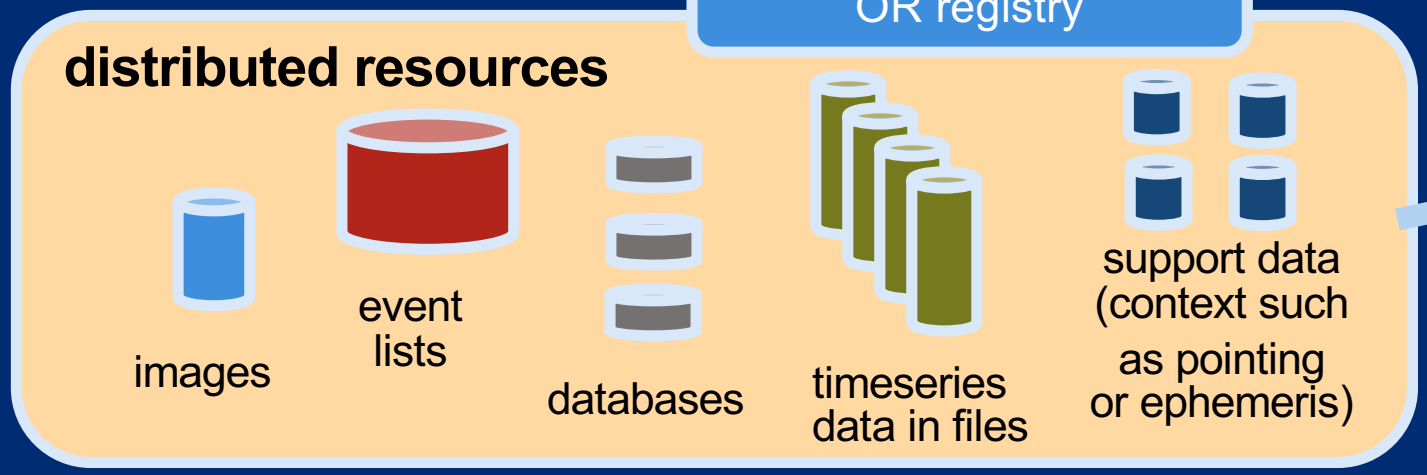
2. results = matching resources
results will vary depending on the system (links, time range only, data files, data stream)

3. analyze resources
plot / filter / save data OR send it to another service

standardized query

pile of resources

may need to go back to the data sources to retrieve content



Many different ways / levels of implementing this.
Examples:
VESPA and SAMP tools
Heliophysics Data Portal and "Get Data" option.

Details about the HAPI data stream format

- time column is first column
  - time format is ISO8601 string (YMD or Year-DayOfYear formats)
  - Examples: 2018-11-10T16:22:00Z 2018-314T16:22:00Z
- data parameters are in subsequent columns
- data types are: isotime, string, double, integer or N-dimensional arrays of these
- you must specify a FILL value or say there is none
- data parameter names must be alpha-numeric (underscores OK)
- for multi-dimensional data, the "spectral bin parameters" can be specified

- CSV format is the only required format for all servers
- optional other formats are: JSON and binary

- data volume limit for a request is up to the server
- there is a standard set of server error codes to implement

HAPI Request / Response Example

http://datashop.elasticbeanstalk.com/hapi/**capabilities**

```
{
  "HAPI": "2.0",
  "status": { "code": 1200, "message": "OK"},
  "outputFormats": [ "csv", "binary", "json" ]
}
```

capabilities

Which standards to focus on now?

3 IVOA Architecture Level 2

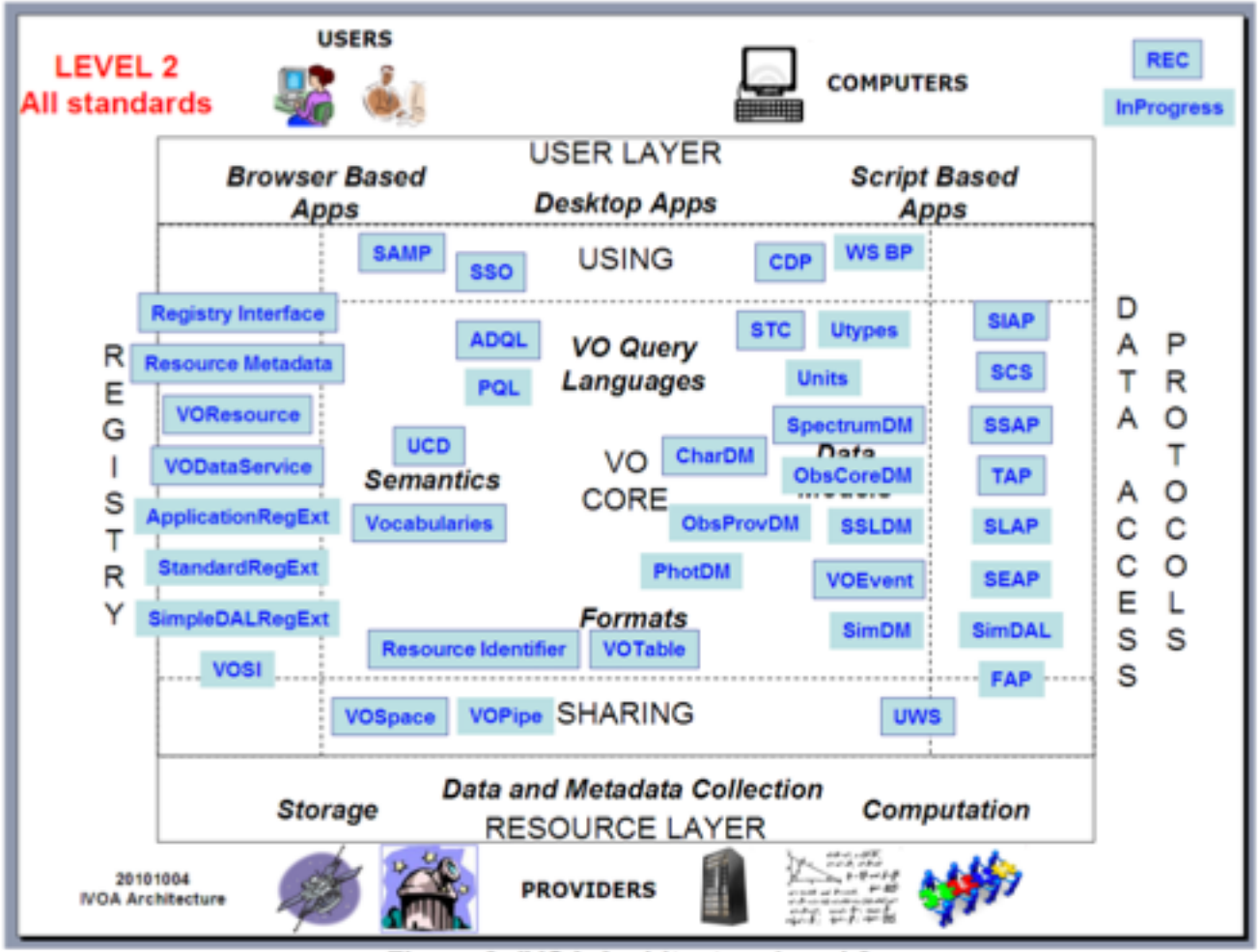


Figure 3: IVOA Architecture Level 2