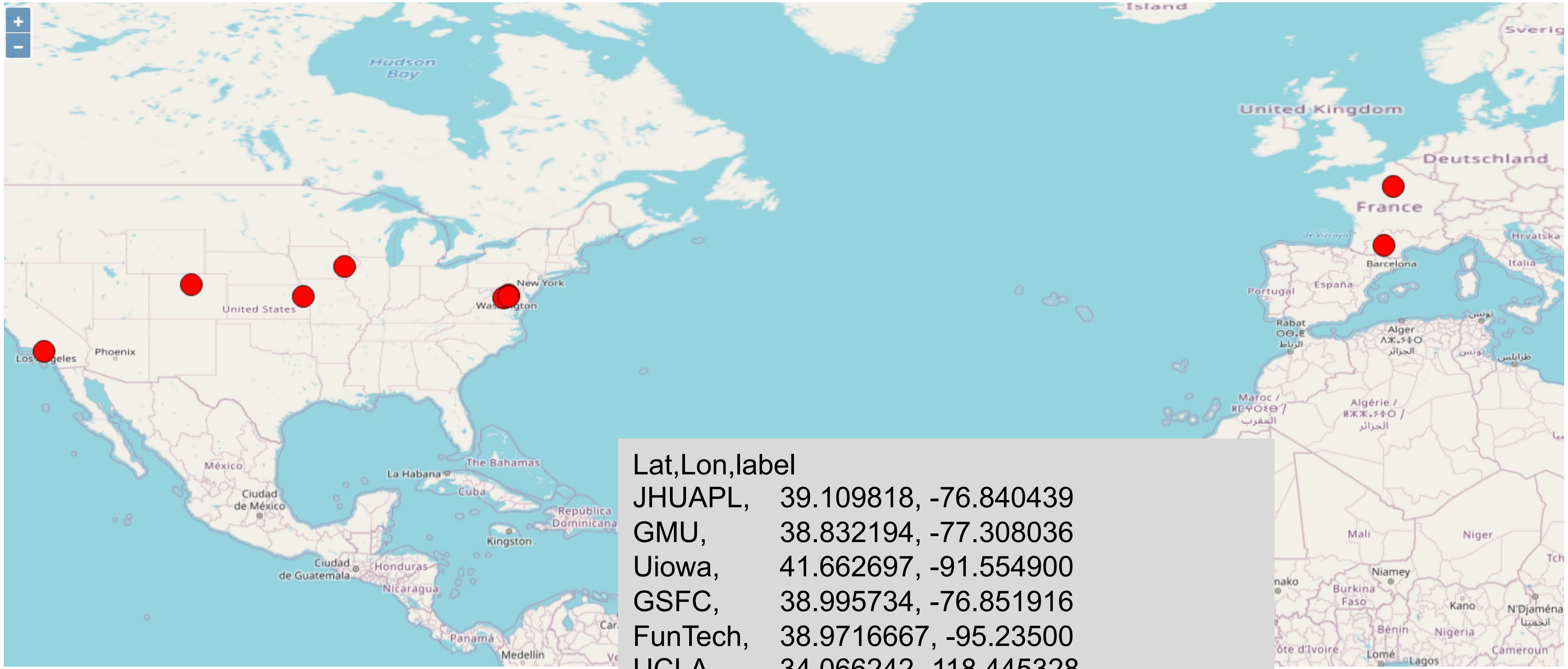
JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

# Using HAPI to enable uniform access to Space Weather timeseries data

**Jon Vandegriff**, Johns Hopkins Applied Physics Lab
**Robert Weigel**, George Mason University
**Jeremy Faden**, Cottage Systems and University of Iowa
**Todd King**, University of California Los Angeles
**D 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**, University of Colorado, LASP
**Chris Lindholm**, University of Colorado, LASP
**Thomas Baltzer**, University of Colorado, LASP
**Larry Brown**, Johns Hopkins Applied Physics Lab
**Eric Grimes**, University of California Los Angeles
**Baptiste Cecconi**, Observatoire de Paris/PSL
**Vincent Génot**, IRAP, CNRS-CNES-UPS, Toulouse, France
**Benjamin Renard**, AKKA Technologies, Toulouse, France
**Arnaud Masson**, Tpz UK for ESA, ESAC, Madrid, Spain
**Beatriz Martinez**, RHEA Group for ESA, ESAC

# HAPI is being adopted internationally



Lat,Lon,label

| | |
|---|---|
| JHUAPL, | 39.109818, -76.840439 |
| GMU, | 38.832194, -77.308036 |
| Uiowa, | 41.662697, -91.554900 |
| GSFC, | 38.995734, -76.851916 |
| FunTech, | 38.9716667, -95.23500 |
| UCLA, | 34.066242,-118.445328 |
| LASP, | 40.014984,-105.270546 |
| Obs de Paris, | 48.836439,2.336506 |
| IRAP, | 43.563151,1.475333 |
| ESAC, | 40.444144,-3.953284 |

NASA

UCLA

CU LASP
Laboratory for Atmospheric and Space Physics

APL JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

THE UNIVERSITY OF IOWA

GEORGE MASON UNIVERSITY

l'Observatoire de Paris

esa

# Motivating the need for interoperability

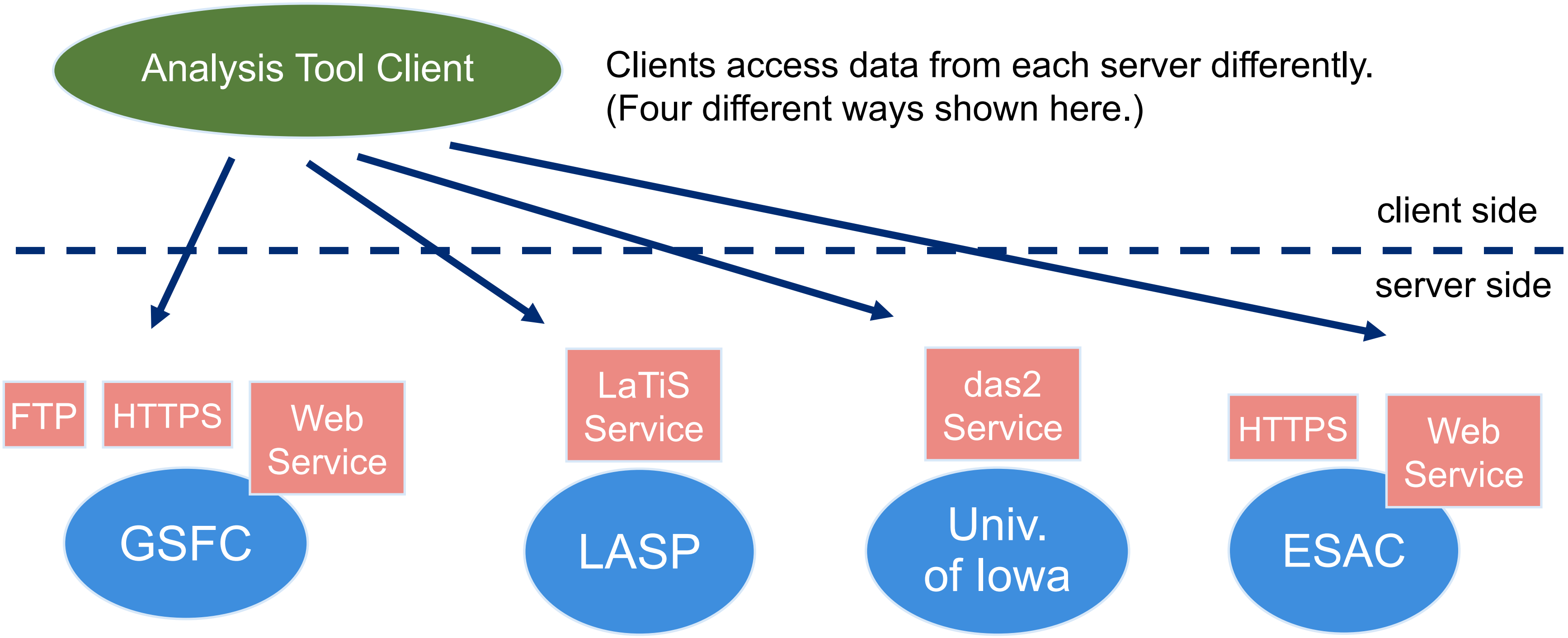
- lots of data sources needed for Space Weather!
  - model input (indices, solar wind properties), model output, spacecraft trajectories, sensor measurements (for comparison with models)

- these usually come from disparate sources

- example: Omniweb data as input to a magnetic field model, whose output should be compared to GOES magnetometer data along the spacecraft trajectory
  - all of these could have a different access method and format

# Putting data online is easy, so diverse options exist

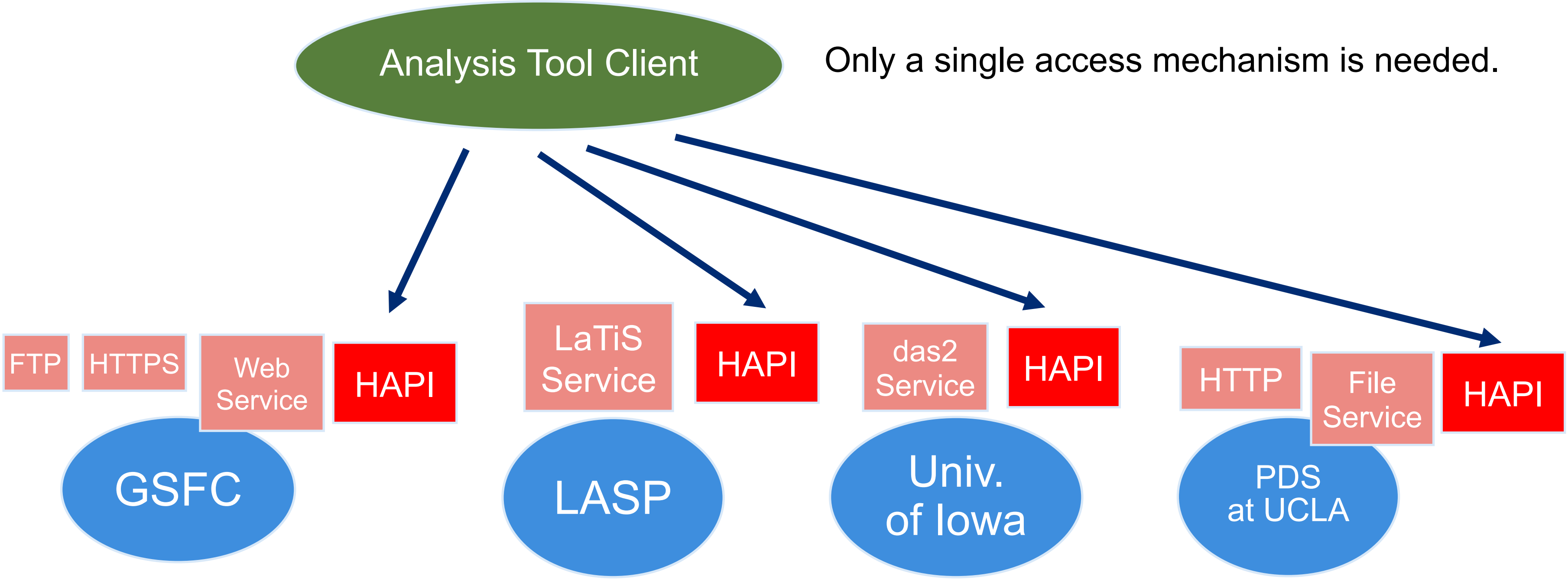
- creating an online data access mechanism has been done many times
- different groups have made similar but not identical access mechanisms
- even if everyone had used the same standard access mechanism, these standards are flexible enough that they would not automatically support interoperability



Clients access data from each server differently.
(Four different ways shown here.)

# Interoperability requires a simple, common access mechanism: HAPI

- nothing new in terms of technology or complexity
- just need everyone to agree to use the same lowest common denominator
- can add a HAPI mechanism alongside existing services



Only a single access mechanism is needed.

# Focus is Timeseries

- conceptually a table, like a spreadsheet
- time column with any number of variables (data columns)
- each variable can be multidimensional

| 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] |

# Existing Types of Data Access (non-HAPI)

1. **direct access to files: http or ftp**
   - obvious standards needed: file format, file names, directory structure
   - more subtle standards needed for full interoperability:
     - file metadata
     - variable layout
     - time value format
2. **service-based access: CGI, web services (RESTful or custom)**
   - obvious standards needed:
     - request format (how to ask for data)
     - response format (what does the returned data look like?)
       - collection of files, OR
       - stream of numbers (JSON, XML, CSV binary, etc)
   - more subtle standards:
     - file metadata
     - variable layout
     - time value format

# HAPI
# Heliophysics Application Programmer's Interface

- first and foremost it is a specification – a server API to make your data interoperable

    main page for all-things HAPI: `https://hapi-server.github.io/`

- two aspects standardized
    - request interface
    - response format

- **key standardizations includes a common format for the time values**
    - subset of ISO 8601 standard for time as a string
    - same format for time values on input and output
    - examples: `2009-315T15:39:30.500Z` `2021-03-17` `2025-01-01T12:00:00Z`

# The request interface: 5 HAPI Endpoints

All endpoints must be directly below a URL that ends with ‘**`hapi`**’

- `http://example.com/hapi/`**`about`**
  - contact info for technical issues and description of server
- `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

*small JSON response*

*possibly large response:*
***CSV** (required)*
***JSON**(optional)*
***binary** (optional)*

# Working examples

## Servers

George Mason
GSFC
Univ. of Iowa
JHU/APL
AMDA

Provisional / Coming Soon:
ESAC
CCMC
LASP
PDS / PPI Node at UCLA

Try this interactive list:
`http://hapi-server.org/servers/`

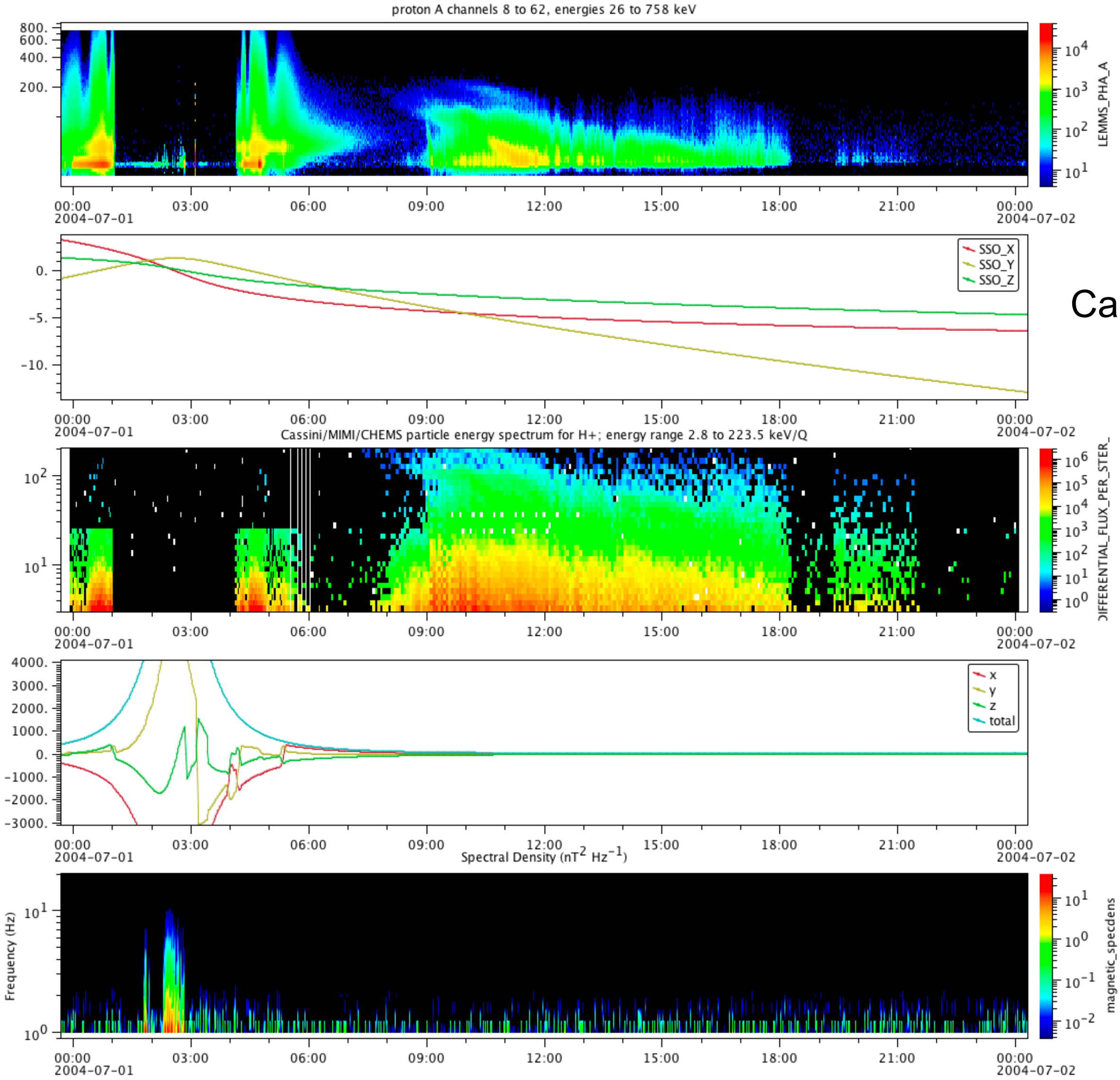
## Clients

Python client
IDL client
Matlab Client

Applications:
-------------------
Autoplot
SPEDAS
PySPEDAS
MIDL4

Client codes as open-source projects:
`https://github.com/hapi-server`

# Autoplot with panels from several HAPI servers



energetic particles (LEMMS proton energy spectra)

Cassini trajectory data

energetic particles (CHEMS proton energy spectra)

MAG data

plasma wave data RPWS

# If you want to serve data using HAPI

- read the specification -- it is very complete
- ways to get to HAPI compliance:
  - modify your existing access mechanism codebase to add HAPI protocols

  -OR-

  - use an existing front end server for managing HAPI details

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

    This is a very well-documented and tested code that can get you up and running quickly!

These two have proven to be very useful!

- For either option above, test your server using this extremely useful verifier tool
  - **`http://tsds.org/verify-hapi`**
    - this checks all of the important parts of the spec, and many corner cases
  - can also test your server in an existing client (Python, 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

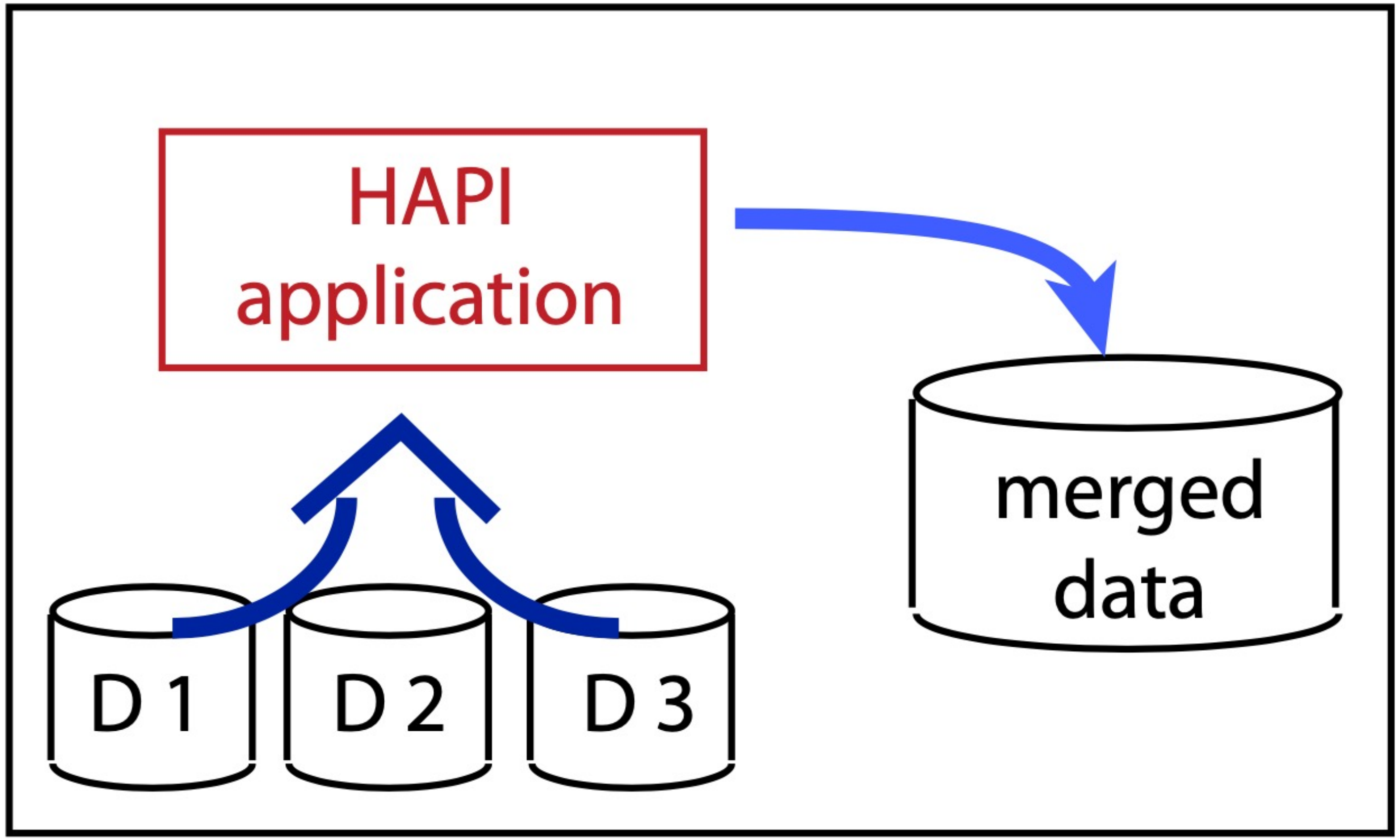
# New and Coming Soon or Eventually

- AMDA HAPI server is up and running
- ESAC HAPI server
  - inevitable, but takes time
  - start with Node.js front end, eventually migrate to native HAPI implementation

- HAPI 3.0 specification – final edits being made
- Some changes in 3.0:
  - regularizing some keywords in the actual API (not backward compatible)
  - time varying bins
  - references in JSON metadata for repeated quantities
  - new about endpoint
- HAPI paper being written
- More focused promotion and support for NASA projects
  - add higher level view – "What HAPI servers are out there?" "Are they up right now?" ”What data do they have?"
  - integration of HAPI within SpacePy and SunPy
  - support for instrument teams – help them use HAPI on their native data systems
  - improve the generic server capability (Python and Java full servers)
  - **add layers on top of HAPI for data mining**

# HAPI and Data Mining

A Python-based, data fusing application is now not too difficult.



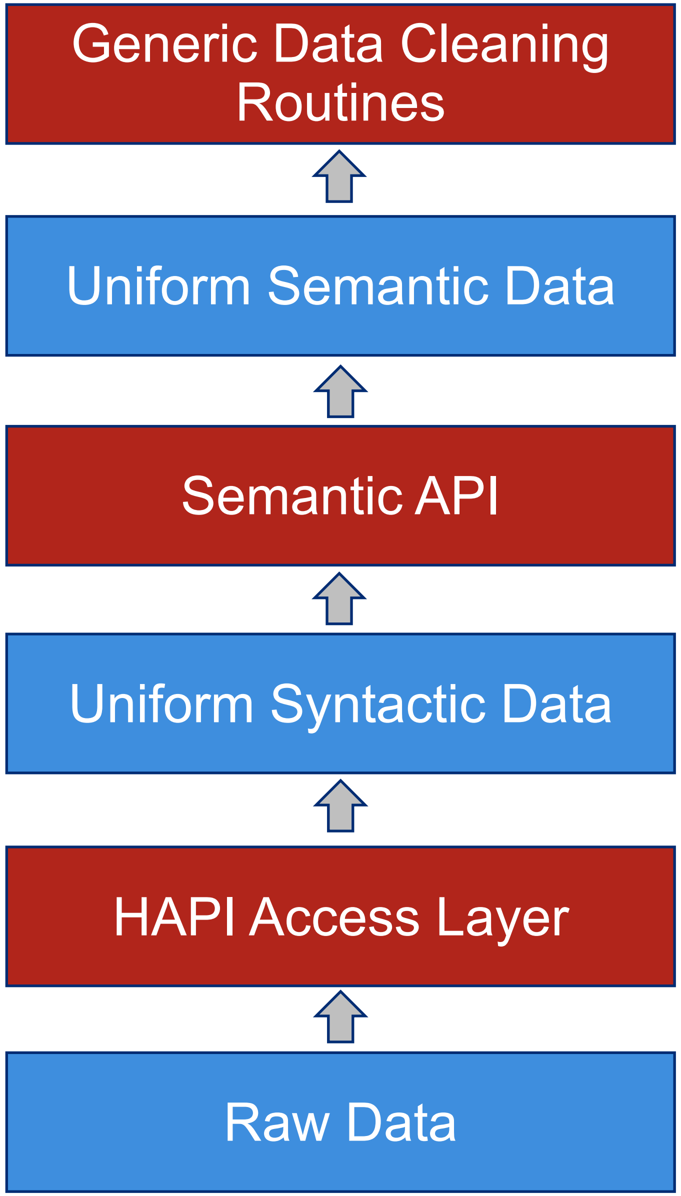
Using the uniformity afforded by HAPI, precursor steps to Machine Learning (access, cleaning, regularizing) can be simplified.

Would be possible to use HAPI in a cloud-based environment (streaming access).

# Next-Level of Uniform Access Beyond HAPI

- HAPI defines an interoperable way to get to the numbers
- Still lots of variability between datasets with similar measurements
- Can we define science-specific, semantically identical, machine readable interfaces to data based on the type of science measurements => Science Data Interfaces
- Create generic API that captures the lowest common denominator of every measurement type:
  - magnetic field
  - plasma wave data
  - plasma moments
  - energetic particles
  - energetic neutral atoms
- Also need similar APIs for numeric ancillary info:
  - spacecraft ephemeris
  - coordinate frames for vectors:
    - spacecraft position (it's a vector)
    - magnetic field direction
    - plasma velocity
    - particle flow direction

# Layers of Interoperability



Generic Data Cleaning Routines

Uniform Semantic Data

Semantic API

Uniform Syntactic Data

HAPI Access Layer

Raw Data

APL

JOHNS HOPKINS

APPLIED PHYSICS LABORATORY