Status of HAPI servers for CDAWeb, SOAR, WDC, and SuperMAG

Jeremy Faden, Cottage Systems
Bob Weigel, George Mason University
Sandy Antunes, Applied Physics Laboratory
Adam Emsley, British Geological Survey

Talk Overview

- Introduce HAPI, Heliophysics Application Programming Interface
- Current HAPI Server at CDAWeb and its issues
- Server-Java framework for building HAPI servers
- New CDAWeb HAPI Server
- Upcoming SOAR HAPI Server at ESAC
- Status of upcoming SuperMag and WDC Servers
- New Caching feature

Introduce HAPI

- Many server types serve time-series data, each needing specialized client software to communicate.
- With Autoplot, I’ll spend months working with server teams to add support for particular server types like CDAWeb, PDS-PPI’s DITDOS, U. Iowa’s Das2Servers, and TAP.
- HAPI tries to identify the common parts and allow services to implement this simple and well-documented API.
- Once HAPI is added, then many existing HAPI clients can immediately access the server.

Introduce HAPI

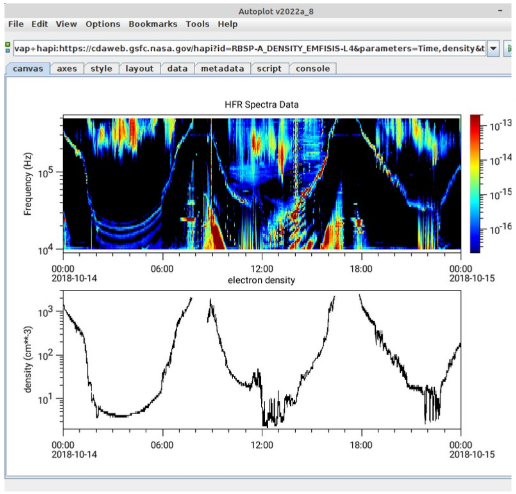
- HAPI was started around 2017 and is a very active project.
- There are some challenging design goals
  - fully developed clients providing rich features like data discovery and time-varying spectrograms
  - to simple wget/curl calls to just "read my data" into CSV file
  - And function within JavaScript containers to provide in-browser clients
- Teams can implement HAPI interface, and software like Autoplot, TOPCAT, and SPEDAS can support them with no new code.
- We provide clients in many languages: Java, JavaScript, Python, IDL, Matlab

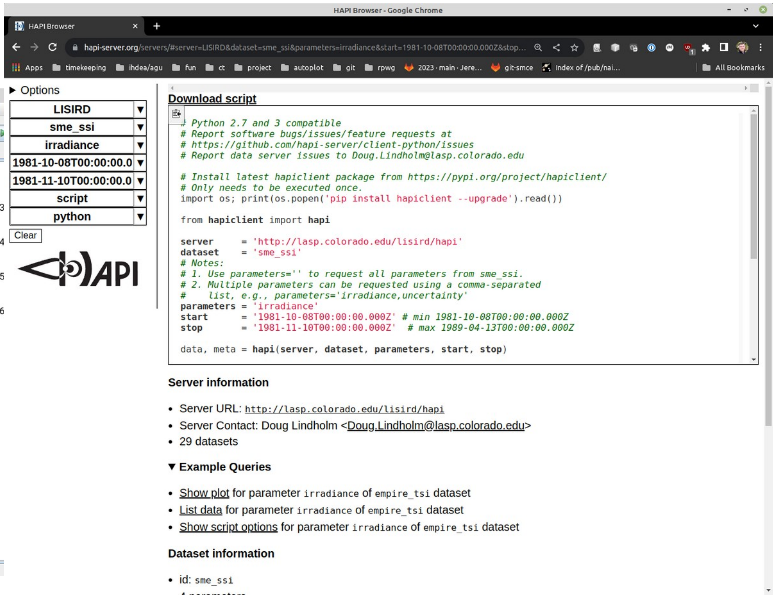
Introduce HAPI

- "catalog" request allows datasets to be discovered. This is a JSON response listing what is available at this site.
- "info" requests describe the dataset for a given id, specifying its labels and units.
- "data" requests are CSV, and optional Binary and JSON responses.
- Streaming is another important aspect of HAPI
  - This means that as data is processed, old resources are closed as new resources are opened.
  - Since the HAPI server streams its data, the scientist can request many years of a mission. The HAPI server is able service this request, it would just take a long time to transfer the data.

Introduce HAPI

- This has been a very active project since 2017, with servers and software written outside of our team. (Of course anybody with time and interest is welcome to join our team!)


Autoplot v2022a_8
File Edit View Options Bookmarks Tools Help
vap+hapi:https://cdaweb.gsfc.nasa.gov/hapi?id=RBSP-A_DENSITY_EMFISIS-L4&parameters=Time,density&t
canvas axes style layout data metadata script console
HFR Spectra Data
Frequency (Hz)
electron density
density (cm**-3)
00:00 2018-10-14
06:00
12:00
18:00
00:00 2018-10-15


HAPI Browser - Google Chrome
hapi-server.org/servers/#server=LISIRD&dataset=sme_ssi&parameters=irradiance&start=1981-10-08T00:00:00.000Z&stop...
Options
LISIRD
sme_ssi
irradiance
1981-10-08T00:00:00.0
1981-11-10T00:00:00.0
script
python
Clear

Download script

```
# Python 2.7 and 3 compatible
# Report software bugs/issues/feature requests at
# https://github.com/hapi-server/client-python/issues
# Report data server issues to Doug.Lindholm@lasp.colorado.edu

# Install latest hapiclient package from https://pypi.org/project/hapiclient/
# Only needs to be executed once.
import os; print(os.popen('pip install hapiclient --upgrade').read())

from hapiclient import hapi

server     = 'http://lasp.colorado.edu/lisird/hapi'
dataset    = 'sme_ssi'
# Notes:
# 1. Use parameters='' to request all parameters from sme_ssi.
# 2. Multiple parameters can be requested using a comma-separated
#    list, e.g., parameters='irradiance,uncertainty'
parameters = 'irradiance'
start      = '1981-10-08T00:00:00.000Z' # min 1981-10-08T00:00:00.000Z
stop       = '1981-11-10T00:00:00.000Z'  # max 1989-04-13T00:00:00.000Z

data, meta = hapi(server, dataset, parameters, start, stop)
```

Server information

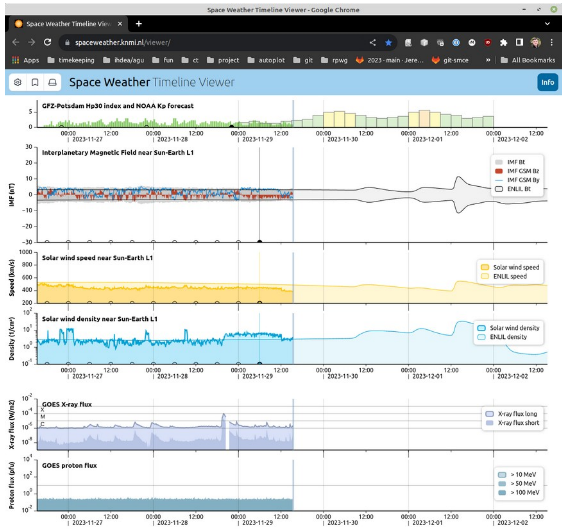
- Server URL: http://lasp.colorado.edu/lisird/hapi
- Server Contact: Doug Lindholm <Doug.Lindholm@lasp.colorado.edu>
- 29 datasets

Example Queries

- Show plot for parameter irradiance of empire_tsi dataset
- List data for parameter irradiance of empire_tsi dataset
- Show script options for parameter irradiance of empire_tsi dataset

Dataset information

- id: sme_ssi



CDAWeb server

Nand Lal wrote the server currently serving data from CDAWeb (roughly 3000 datasets).

He passed away in 2021, suddenly, and I along with Bernie Harris and Jennifer Sun at NASA/SPDF have taken over its maintenance.

- Challenging environment: it's in production use by scientists, but it's also used for testing and receives many hostile attacks as a high-visibility server.
- It is a HAPI version 2.0 server (~2019) doesn't support many CDF features, like time-varying dependencies and metadata. Later HAPI versions do support this.
- performance is irregular, needing to be restarted often.
- start-up is slow, about 45 minutes, indexing all datasets and identifying HAPI responses.
- Response cache added to old server, partially resolving the slow startup time.

With all these issues it is clear that a replacement for the server is needed

CDAWeb server - Deep Dive

- Last failure back in August, I decided to take a deep dive into the code to understand it better.
- It looked like he was introducing multi-threaded code which would load the next CDF into memory while the first was being sent out.
- I found the bug, where mistake in this code meant all data timetags were kept in memory, never cleared after they were transmitted.
- A month-long request of MMS burst data would fill up memory and crash the server.
- This has been fixed, and a new revision of the server is running at CDAWeb.

CDAWeb – New server

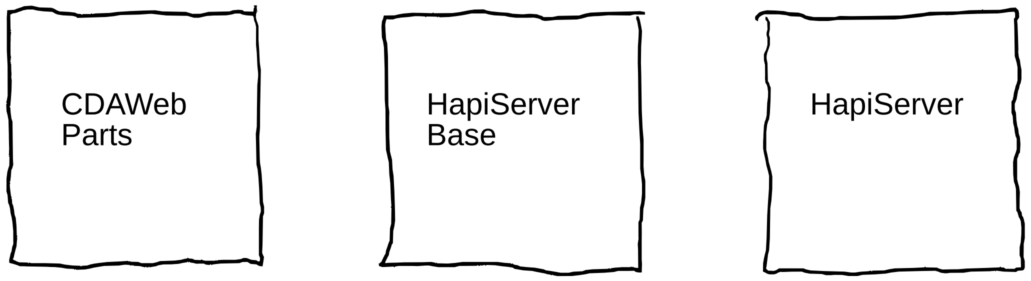
- Bob Weigel and I have been working on a replacement server for CDAWeb.
- start-up time reduced to seconds
- HAPI version 3.2 server will fully describe CDF data
  - time-varying DEPEND_1
  - coordinate frames
  - we did not use SPASE metadata (tried, however), will suggest improvements for SPASE
- Performance improvements because of CDF_TT2000 (double) to ISO8601 time formatting. New code can format timetags 5-10 times faster.

CDAWeb server - new

- Based on the "Server-Java" framework
  - separates the mundane HAPI parts like trimming responses from the code needed to implement a specific server.
  - provides caching of parts of the server for info and catalog responses
  - framework makes maintenance much easier for me and others.
    - The old server is about 8000 lines of code in 74 files,
    - new server is about 2500 lines of code in 20 files.

server-java

- Instead of one monolithic code which implements a server, we have separate parts for:
  - the interface to a specific data source like CDAWeb,
  - the "HAPI parts",
  - the web server interface



Implements with minimal code

Verifies and formats HAPI responses

Uses HapiServer Base to implement web server

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

server-java - HapiServerBase

- subset the data in time and by parameter
- format to CSV, Binary, or JSON
- validation
  - is the stream sorted?
  - do data columns match the info specified?
- utility functions like formatting times efficiently
- streaming may be done file-by-file. Read a file, stream the data, repeat...

server-java - specific server

- boolean hasGranuleIterator()
  - for the CDAWeb case, this is true, one granule for each CDF file
- Iterator<TimeRange> getGranuleIterator(Time start, Time stop)
  - for the CDAWeb case, this corresponds to the time spans each file covering the interval
- boolean hasParamSubsetIterator()
  - For CDAWeb, this is true, we only want to provide the parameters requested, not all parameters
- Iterator<HapiRecord> getIterator(Time start, Time stop, String[] params)
  - returns an iterator for the HapiRecords
  - Start, stop times will be the range returned by getGranuleIterator

CDF Files on a disk

c3_pp_cis_20050912_v01.cdf c3_pp_cis_20050913_v01.cdf c3_pp_cis_20050914_v01.cdf c3_pp_cis_20050915_v01.cdf c3_pp_cis_20050916_v01.cdf

```
2005-09-12T00:00:21.948Z,0.010...
2005-09-12T00:00:26.070Z,0.007...
2005-09-12T00:00:34.326Z,0.003...
2005-09-12T00:01:07.318Z,0.010...
2005-09-12T00:01:15.576Z,0.003...
2005-09-12T00:01:32.078Z,0.070...
2005-09-12T00:01:40.322Z,0.007...
2005-09-12T00:01:44.456Z,0.005...
2005-09-12T00:02:13.330Z,0.005...
2005-09-12T00:02:25.698Z,0.003...
2005-09-12T00:02:29.818Z,0.005...
2005-09-12T00:02:58.702Z,0.015...
2005-09-12T00:03:15.206Z,0.009...
2005-09-12T00:03:35.830Z,0.004...
2005-09-12T00:03:52.336Z,0.011...
2005-09-12T00:04:08.822Z,0.002...
2005-09-12T00:04:25.326Z,0.013...
2005-09-12T00:04:37.706Z,0.010...
2005-09-12T00:05:14.834Z,0.006...
2005-09-12T00:05:18.956Z,0.007...
2005-09-12T00:05:23.078Z,0.011...
2005-09-12T00:06:49.718Z,0.007...
2005-09-12T00:07:22.722Z,0.008...
2005-09-12T00:07:51.588Z,0.008...
...
```

```
2005-09-13T00:29:44.338Z,3.152...
2005-09-13T09:57:13.206Z,0.011...
2005-09-13T10:00:47.718Z,0.004...
2005-09-13T11:54:51.382Z,0.002...
2005-09-13T12:21:31.952Z,0.005...
2005-09-13T12:22:37.948Z,0.003...
2005-09-13T13:27:27.986Z,2.055...
2005-09-13T15:55:54.280Z,0.004...
2005-09-13T15:57:53.910Z,0.005...
2005-09-13T15:59:45.292Z,0.006...
2005-09-13T16:03:52.808Z,0.004...
2005-09-13T16:04:01.050Z,0.004...
2005-09-13T16:07:23.178Z,0.114...
2005-09-13T16:08:00.308Z,0.123...
2005-09-13T16:08:04.444Z,0.069...
2005-09-13T16:10:32.952Z,0.010...
2005-09-13T16:10:53.578Z,0.032...
2005-09-13T16:11:51.316Z,0.087...
2005-09-13T16:11:55.440Z,0.081...
2005-09-13T16:12:07.824Z,0.110...
2005-09-13T16:12:11.944Z,0.110...
2005-09-13T16:12:24.324Z,0.126...
2005-09-13T16:12:32.570Z,0.134...
2005-09-13T16:12:44.948Z,0.103...
...
```

```
2005-09-14T01:18:35.060Z,6.235...
2005-09-14T04:06:12.352Z,0.002...
2005-09-14T04:19:24.396Z,0.002...
2005-09-14T04:24:13.162Z,2.809...
2005-09-14T04:45:07.246Z,0.004...
2005-09-14T04:47:39.880Z,8.839...
2005-09-14T05:30:34.028Z,0.038...
2005-09-14T05:44:35.578Z,0.001...
2005-09-14T05:52:50.610Z,0.085...
2005-09-14T06:17:56.330Z,0.028...
2005-09-14T06:18:04.572Z,0.010...
2005-09-14T06:18:16.952Z,0.013...
2005-09-14T06:18:37.580Z,0.005...
2005-09-14T06:20:08.338Z,0.005...
2005-09-14T06:20:45.468Z,0.010...
2005-09-14T06:25:13.608Z,0.003...
2005-09-14T06:36:01.258Z,0.030...
2005-09-14T06:42:04.282Z,0.016...
2005-09-14T06:51:04.702Z,1.576...
2005-09-14T07:27:31.076Z,0.006...
2005-09-14T07:39:24.742Z,0.012...
2005-09-14T07:40:34.874Z,0.011...
2005-09-14T07:40:43.132Z,0.004...
2005-09-14T07:40:55.500Z,0.006...
...
```

```
2005-09-15T00:01:00.178Z,0.012...
2005-09-15T00:02:18.558Z,0.007...
2005-09-15T00:02:30.940Z,0.010...
2005-09-15T00:02:35.062Z,0.027...
2005-09-15T00:02:39.182Z,0.016...
2005-09-15T00:02:43.308Z,0.008...
2005-09-15T00:02:47.428Z,0.006...
2005-09-15T00:02:55.686Z,0.009...
2005-09-15T00:03:49.320Z,0.010...
2005-09-15T00:05:57.190Z,0.027...
2005-09-15T00:06:09.570Z,0.012...
2005-09-15T00:06:21.954Z,0.161...
2005-09-15T00:07:15.570Z,0.014...
2005-09-15T00:07:32.074Z,0.024...
2005-09-15T00:07:44.456Z,0.013...
2005-09-15T00:07:56.836Z,0.019...
2005-09-15T00:08:50.452Z,0.013...
2005-09-15T00:10:25.332Z,0.014...
2005-09-15T00:12:53.844Z,0.007...
2005-09-15T00:13:35.094Z,0.006...
2005-09-15T00:14:28.724Z,0.006...
2005-09-15T00:15:26.480Z,0.004...
2005-09-15T00:15:42.984Z,0.005...
2005-09-15T00:17:17.864Z,0.001...
...
```

```
2005-09-16T00:06:12.638Z,0.006...
2005-09-16T00:20:30.678Z,0.003...
2005-09-16T00:21:11.928Z,0.003...
2005-09-16T01:24:51.900Z,0.014...
2005-09-16T01:37:06.186Z,0.014...
2005-09-16T01:40:28.330Z,0.047...
2005-09-16T01:40:36.576Z,0.037...
2005-09-16T01:43:42.214Z,0.026...
2005-09-16T01:45:17.098Z,0.026...
2005-09-16T01:45:21.214Z,0.028...
2005-09-16T01:45:41.844Z,0.036...
2005-09-16T01:46:14.846Z,0.002...
2005-09-16T01:47:41.472Z,0.011...
2005-09-16T01:49:12.230Z,0.024...
2005-09-16T01:49:49.358Z,0.007...
2005-09-16T01:54:05.120Z,0.024...
2005-09-16T01:56:54.254Z,0.006...
2005-09-16T02:07:00.672Z,1.850...
2005-09-16T02:12:38.932Z,0.008...
2005-09-16T02:14:13.814Z,0.014...
2005-09-16T02:14:42.702Z,0.017...
2005-09-16T02:14:55.064Z,0.009...
2005-09-16T02:15:52.830Z,0.010...
2005-09-16T02:19:56.214Z,0.009...
...
```

Each read into stream of HapiRecord by the specific server code and written as CSV to output by HapiServerBase

server-java - planned features

- multi-threaded: so data can be transmitted as the next CDF file is read in
- The code specific to CDAWeb wouldn't need to change, as the HapiServerBase portion would handle this.
- This will improve performance for all servers based on the server-java framework.

SOAR Server at ESAC

- Built on the same "Server Java" framework as the CDAWeb server and Cluster Science Archive HAPI server.
- Data is in granules, which are CDF files listed by SOAR TAP service
  - https://soar.esac.esa.int/soar-sl-tap/tap/sync?REQUEST=doQuery...
- Each of the CDF files is accessed serially and records are provided to the server using the same code that the CDAWeb server uses.
- Jonathan Cook and I will be working on it this week, and we plan to put it into production over the next few months.
- We expect more datasets will be added to the server after it goes into production.

WDC server

- Adam Emsley at the British Geological Survey is the lead developer.
- They have ground magnetometer data going back to the 1960s.
- They had developed a custom API and we asked them to consider HAPI.
- They were enthusiastic and had a prototype in a week. It is nearly in production at https://wdcapi.bgs.ac.uk/hapi
- Many discussions on metadata, especially regarding conveying provenance and being consistent with INTERMAGNET.
- There is much overlap in magnetometer data available from INTERMAGNET, SuperMag, and WDC. There is a need for a way for users to be able to search all. Several ways to do this - one is to auto create SPASE from their HAPI servers. Another is to ask them to create SPASE and post to hpde.io

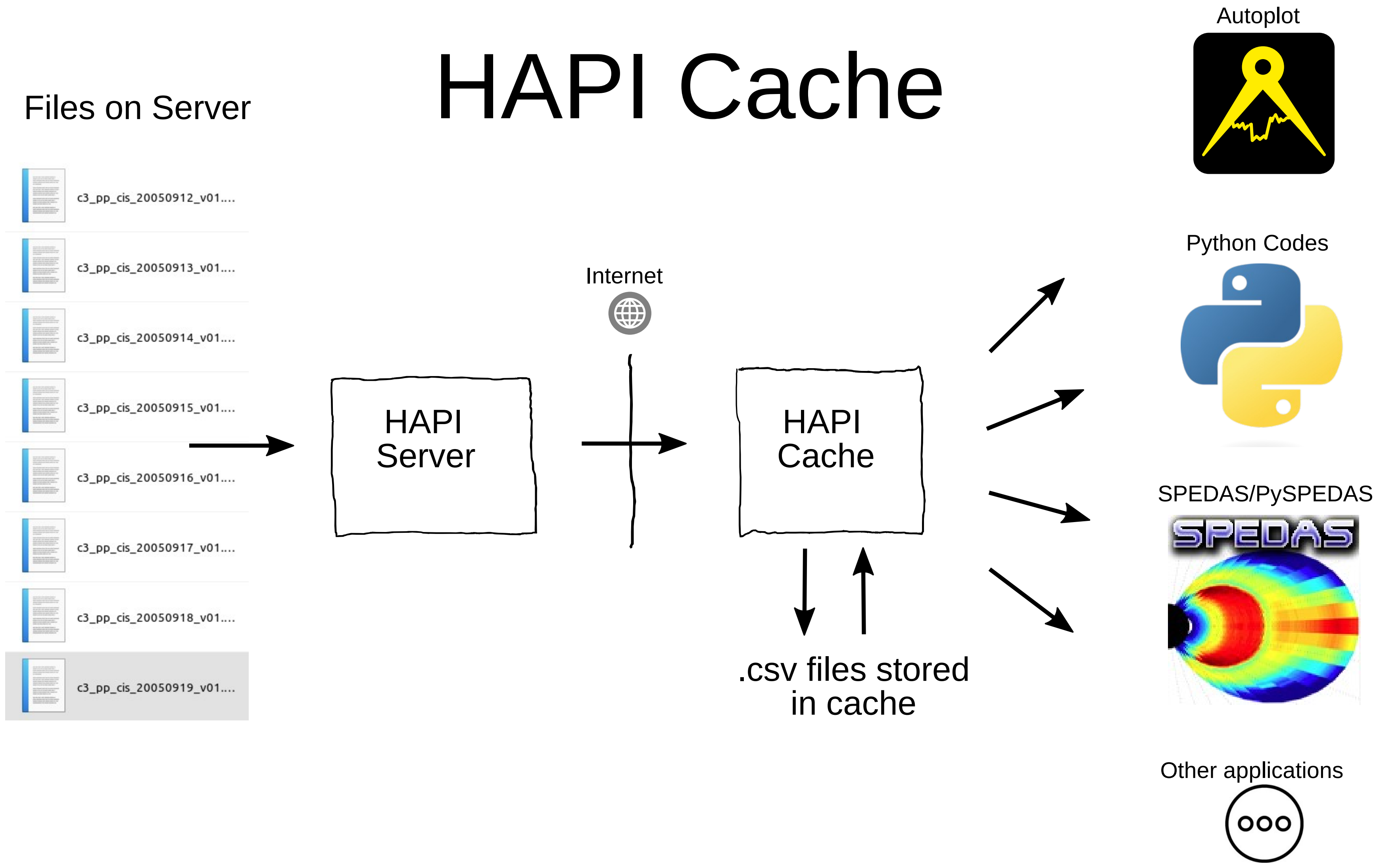
SuperMag server

- Several attempts over the past six years to get one running.
- Was easy to create a pass-through server, but it could not make it public due to their data use policy and captchas. Captcha became non-issue when they released client software, but data policy prevented the release of pass-through server.

HAPI Cache

- A problem with web services like HAPI is that when the server is not accessible, its clients (often scientists) are unable to work.
- Beyond that, you spend an awful amount of time downloading the same data over and over again.
- Usually you want the latest data, but you also might keep a particular version of the data for a study.
- New cache software will work with all HAPI clients, so you can start your study in Autoplot and then do analysis in Python offline.
- Data is kept in one location, and cache will have options to control freshness.
- This will be presented at AGU on Wednesday AM, IN31C.

HAPI Cache


Files on Server
c3_pp_cis_20050912_v01....
c3_pp_cis_20050913_v01....
c3_pp_cis_20050914_v01....
c3_pp_cis_20050915_v01....
c3_pp_cis_20050916_v01....
c3_pp_cis_20050917_v01....
c3_pp_cis_20050918_v01....
c3_pp_cis_20050919_v01....
HAPI Server
Internet
HAPI Cache
.csv files stored in cache
Autoplot
Python Codes
SPEDAS/PySPEDAS
SPEDAS
Other applications

Summary

- HAPI is a mature protocol which can be added to time series data servers.
- The legacy CDAWeb HAPI server is being replaced with a new HAPI 3.2 version which will have new features.
- A SOAR HAPI server is being set up on top of the TAP service, providing access to more clients. This is built on the same code base as the new CDAWeb HAPI server and Cluster Science Archive server.
- SuperMag server is coming soon.
- WDC server is up and running.
- Caching will support offline use of HAPI servers and other use cases.

Thanks!

Jeremy Faden <faden@cottagesystems.com>