



# A Web Services Interface to Advanced Composition Explorer (ACE) Data

Andrew Davis, Glenn Hamell

Spring AGU, May 2005



- [ASC Home](#)
- [What's New?](#)
- [ACE News](#)
- [Related Sites](#)
- [Meetings](#)

# ACE Level 2 (Verified) Data

[ACE Level 2 Data Policy](#)

- Online Data
- [Level 2 \(Verified\)](#)
- [Real Time](#)
- [Browse](#)
- [Level 1](#)
- [Contributed](#)
- [ACE Orbit](#)
- [Time Cal.](#)
- [Housekeeping](#)

- Documentation
- [Data Docs](#)
- [Mission Docs](#)
- [ACE Instruments](#)
- [Reports/Status](#)
- [Publications](#)

Interplanetary Magnetic Field Parameters: MAG	<a href="#">MAG Data</a>	<a href="#">Documentation</a>
Solar Wind Parameters: SWEPAM	<a href="#">SWEPAM Data</a>	<a href="#">Documentation</a>
Solar Wind Temperatures, Speeds, and Species Ratios: SWICS	<a href="#">SWICS Data</a>	<a href="#">Documentation</a>
Solar Suprathermal and Energetic Particle Intensities: ULEIS	<a href="#">ULEIS Data</a>	<a href="#">Documentation</a>
Solar Energetic Particle Intensities: EPAM	<a href="#">EPAM Data</a>	<a href="#">Documentation</a>
Solar Energetic Particle Intensities: SEPICA	<a href="#">SEPICA Data</a>	<a href="#">Documentation</a>
SEP, GCR, and ACR Intensities: SIS	<a href="#">SIS Data</a>	<a href="#">Documentation</a>
Galactic Cosmic Ray Intensities: CRIS	<a href="#">CRIS Data</a>	<a href="#">Documentation</a>
<b>NEW</b> Merged IMF and Solar Wind 64-second Averages	<a href="#">MAG/SWEPAM Data</a>	<a href="#">Documentation</a>
<b>NEW</b> Merged IMF, Solar Wind, and Energetic Particle Hourly Averages	<a href="#">Multi-instrument Data</a>	<a href="#">Documentation</a>

[Acknowledgement Guidelines](#)

[Data Processing Changelog](#)

[Data Inventory Graph](#)

[Register as ACE Data User](#)



## Define the Problem...

We want to open up ACE Science Center services and data to remote computers and services. However...

- Access to ACE data is currently via a web-based forms interface, designed for human interaction.
- The interface is non-standard, undocumented.
- We do not have resources to build a new interface from scratch – we want to integrate our existing (legacy) interface.
- Current descriptions of ACE data products are also non-standard.

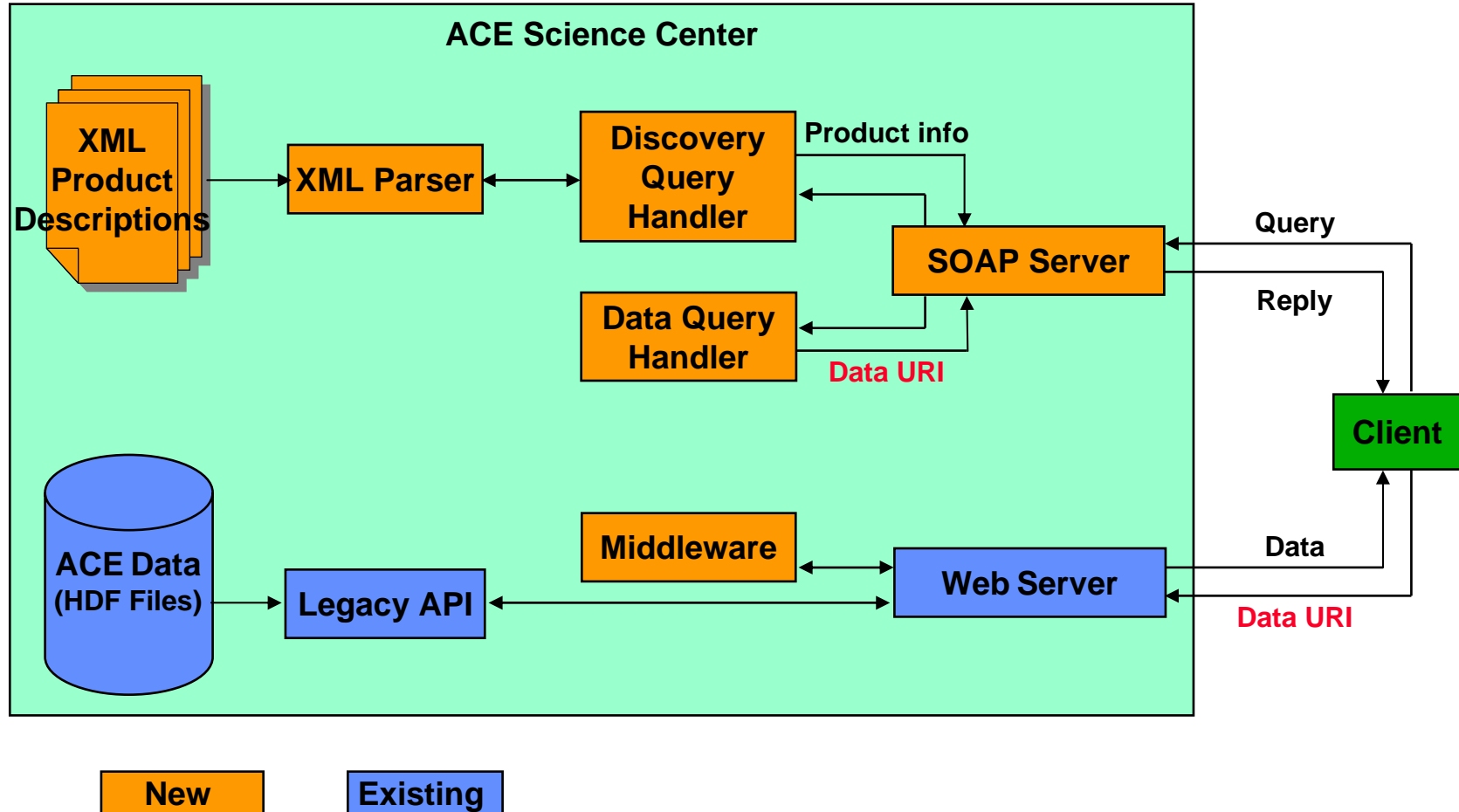


## Proposed Solution...

- Describe ACE data products in XML, using the Space Physics Archive Search and Extract (SPASE) Project Data Model.
- Use Simple Object Access Protocol (SOAP) as the communications protocol to handle Discovery queries and Data queries.
  - Build a SOAP server to handle queries
- Build a middleware that mediates between our legacy interface and the new service.
- Use HTTP protocol for final transfer of data.



# Service Architecture





# SOAP Server

- Implemented in Perl, using SOAP::Lite and XML::Simple modules
- Currently supports the following queries:
  - **Discovery** – Server replies with summary information on each data product available from the ASC
  - **Get\_Details(Product\_ID)** – Server replies with detailed info on each data item for a given data product
  - **Get\_Data(Params)** – Client requests a subset of data; Server replies with a HTTP GET URI to the data subset.
- Easily adaptable to emerging standards... 😊
- Example client application available on request (written in Perl)

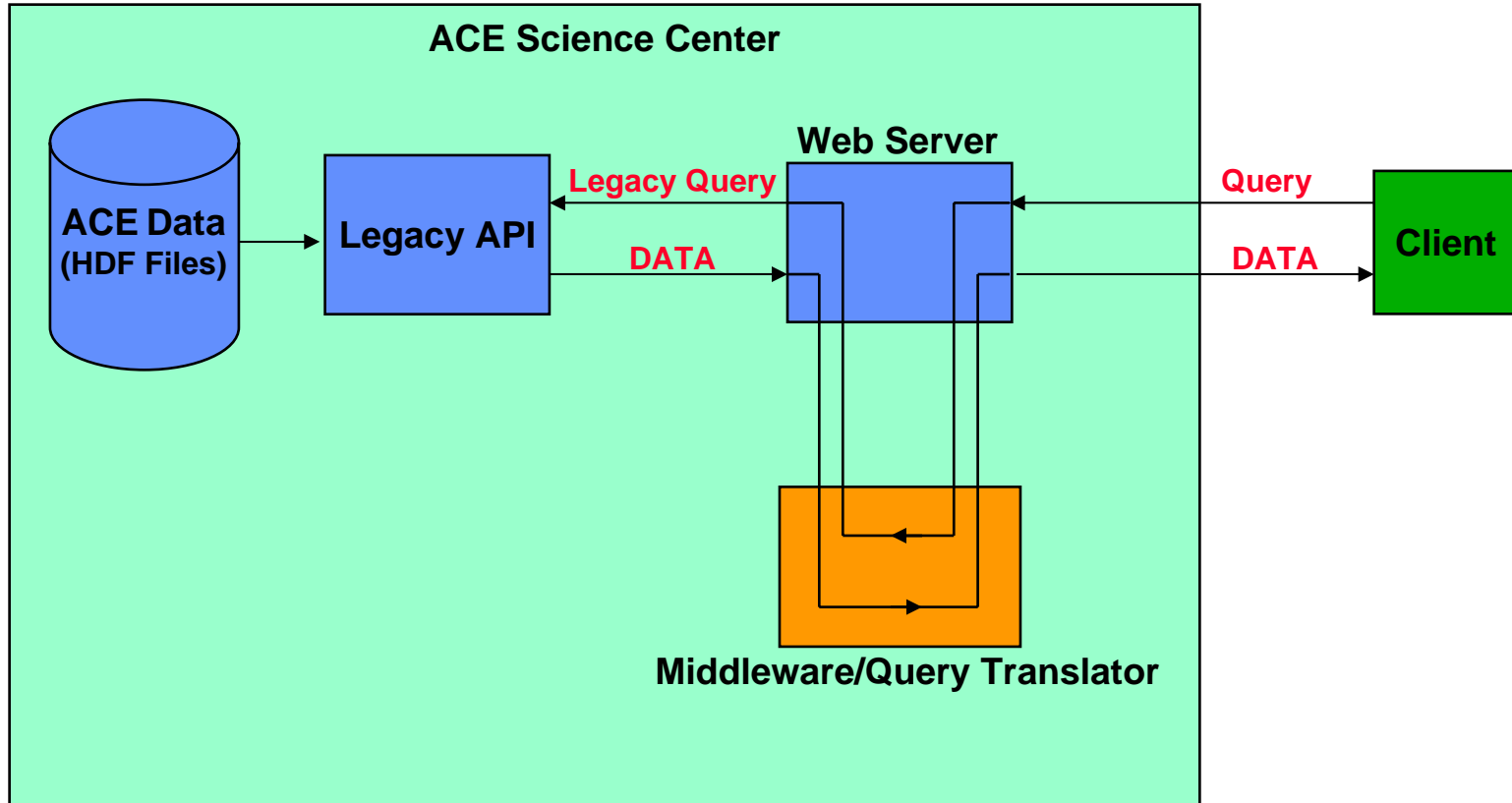


## Middleware

- Mediates between legacy API and new service.
- Accepts simple HTTP GET requests for data subsets.
- Translates requests into the more obtuse HTTP POST requests required by the legacy API.
- Streams ASCII data from the legacy API to the client as HTTP file download (bent pipe).
- Implemented as a Perl CGI script.



# Middleware



- Could simplify by having middleware interface directly to Legacy API.





# XML Product Descriptions: SPASE Based

```
<Product><Header>
  <Product_ID>ASC_ACE_ACE_MAG_ASCII_3600s_v1</Product_ID>
  <Product_Name>ACE MAG Hourly Level 2 Data</Product_Name>
  <Description>Hourly Averaged Interplanetary Magnetic Field Data from ACE/MAG</Description>
  <Repository_Name>ACE Science Center (ASC)</Repository_Name>
  <Project>
    <Observatory>ACE</Observatory> <Instrument_Name>MAG</Instrument_Name>
    <Instrument_Type>Magnetometer</Instrument_Type>
  </Project>
  <Contact>
    <Name> <First>Chuck</First> <Middle>W</Middle> <Last>Smith</Last> </Name>
    <Institution>University of New Hampshire</Institution>
    <Role>Instrument Scientist</Role> <E-mail>Charles.Smith@unh.edu</E-mail>
  </Contact>
  <Product_Type><Numerical_Data>
    <Format>ASCII</Format> <File_Encoding>UNIX (ASCII)</File_Encoding>
  </Numerical_Data></Product_Type>
  <Availability>Online</Availability>
  <Access_Rights>Open</Access_Rights>
  <Measurement_Type>Magnetic Field</Measurement_Type>
  <Processing_Level>Calibrated</Processing_Level>
  <Instrument_Region>L1</Instrument_Region> <Observed_Region>L1</Observed_Region>
  <Resolution>1hr</Resolution>
  ...and so on and so on..
```



## Detailed Time-Series Descriptions...

```
<Time_Series>  
  <Time_Series_Description>  
    GSM X component of hour-averaged magnetic field  
  </Time_Series_Description>  
  <Physical_Quantity>MAG_FIELD</Physical_Quantity>  
  <Entity>FIELD</Entity>  
  <Qualifier>COMPONENT</Qualifier>  
  <Qualifier>X</Qualifier>  
  <Unit>NT</Unit>  
  <Coordinate_System>GSM</Coordinate_System>  
  <Query_Code>Bgsm_x</Query_Code>  
</Time_Series>
```

One of these for each data item in a product...



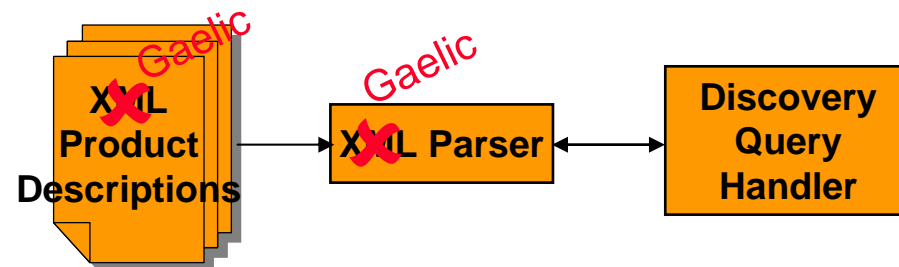
## Current Status

- The service is up and running - beginning to connect to VHO, VSPO, etc.
- Example client application is available on request (Perl script) – **beta-testers needed!**
- Interested in learning about standards for formulating space-physics discovery and data queries using the SOAP protocol.



## Lessons Learned?

- Current toolkits (e.g. Perl's SOAP::Lite) make implementation of a SOAP service relatively painless – not too much prior expertise needed.
- Most important step – getting product metadata into a structured format – not necessarily XML!



- Not a task one could tackle on a single slow afternoon...



## SOAP Server CGI script

```
#!/usr/local/bin/perl -w  
use SOAP::Transport::HTTP;  
  
use Discover;  
use GetDetails;  
use GetData;  
  
SOAP::Transport::HTTP::CGI  
->dispatch_to('Discover','GetDetails','GetData')  
->handle;
```



## Example Data Query

```
$som = $soap->getdata(  
    SOAP::Data->name('DataQuery' =>SOAP::Data->value(  
        SOAP::Data->name("instrument_name" => "MAG"),  
        SOAP::Data->name("resolution" => "1hr"),  
        SOAP::Data->name("data1" => "Brtn_r"),  
        SOAP::Data->name("data2" => "Brtn_t"),  
        SOAP::Data->name("data3" => "Brtn_n"),  
        SOAP::Data->name("start" => "2002 1"),  
        SOAP::Data->name("end" => "2002 2")  
    ));
```