

# FDSN Web Services in Antelope

Rohan Ambli  
Ambli, LLC

May 2019  
Taormina, Sicily





# Outline

- Background
- Web-services Overview
- FDSN web-service (data export)
- Client access (data import)
- Real-time system example
- Feedback



# Background

- 10+ years in the industry
- Automation Engineer, with interest in Kubernetes and Docker
- From India. Living in Denver, Colorado since 2005



# Background

- Ambli, LLC Contracting with BRTT
- Finished Antelope port to Python 3, part of Antelope 5.9 release
- On-going effort: FDSN Web-service





# Me at AUG



Me



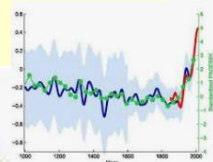
trace  
view



Q330M+



dbloc



preferred  
origin





# Web services

- What
  - A software system designed to support machine to machine interaction over a network.
- Why
  - Interoperability between distributed systems over the internet/intranet



# Web services

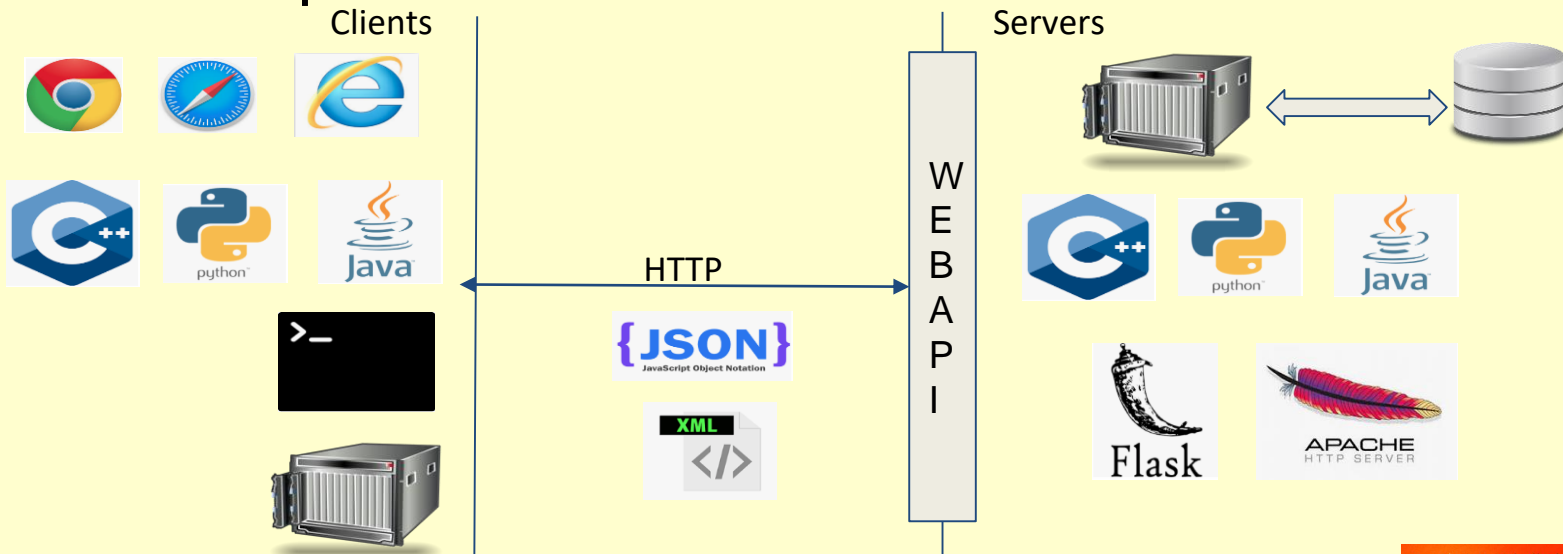
- How
  - Protocol: HTTP (HyperText Transfer Protocol)
  - Well defined interface: REST/SOAP
  - Same language: XML or JSON





# Web services

- De-couple client implementation from server implementation
  - platform independent





# Web services

- Easy data access
- Standardize local data access as well
- Service description information over WADL



# FDSN Web-service

- Growing interest in Federated\* Data Services for Seismology
- *“Members agree to coordinate station siting and provide free and open access to their data”* - [www.fdsn.org](http://www.fdsn.org)
- FDSN web service specification defines RESTful web-service interfaces for accessing common FDSN data types.
- Antelope Prototypes started March, 2019

\* Federated architecture (FA) is an architecture pattern that allows interoperability and information sharing between semi-autonomous de-centrally organized entities



# Federated Metadata Aggregator

## Data Services Newsletter

Volume 21 : No 1 : Spring 2019

◀ Previous: ISPAQ V2.0 - Improvements to the IRIS System for Portable Assessment of Quality

Next: Jerry A Carter - Director of Data Services ▶

🚩 Web Update

## Federated MetaData Aggregator (MDA)

### Introduction

The IRIS MetaData Aggregator has been redesigned to show metadata summaries from global data centers supporting FDSN Web Services. In addition to SEED metadata information, MDA presents other details such as data availability, virtual network affiliations, response curves, and links to dynamic maps. MDA adds to the IRIS DMC's suite of federated web tools including GMAP and Wilber 3, which collectively provides users a convenient view of the station holdings provided by all participating FDSN data centers.

MDA exposes five levels of metadata summary: network list, network, station, location and channel. At each summary level, metadata from different federated datacenters is related using SEED metadata identifiers and time spans.



<http://ds.iris.edu/ds/newsletter/vol21/no1/507/federated-metadata-aggregator-md/>





# FDSN Web-service

`station/` – Returns metadata in FDSN StationXML and alternate formats

`event/` – Return event (earthquake) information

`dataselect/` – Provides access to time-series data for specified channels and date ranges



# Data export and import

- Data *export* is handled by Antelope FDSN web-server
  - `station/` - StationXML export based on *db2stationxml(3)* written by Celso Reyes
  - `event/` - QuakeML export currently based on *event2qml(1)* contrib code written by Juan Reyes
  - `dataselect/` - Current experiments based on *trexcerpt(1)*
- Data *import*\* can be performed by any client supporting REST calls.

- `curl` command
- Web UI
- Obspy Python package

\* *import* here refers to importing data for analysis, not to Antelope db (not *stationXMLtoDB*)



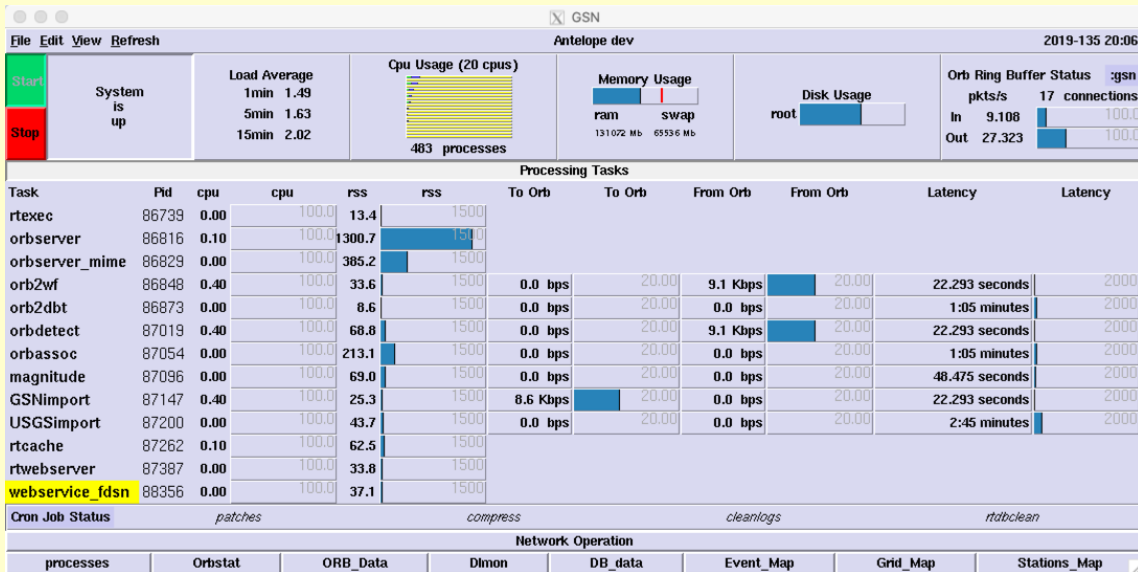
# FDSN Server Export from Antelope - Server within rtdemo\_gsn

New experimental server  
'webservice\_fdsn'

In rtexec.pf:

```
Processes &Tbl{
webservice_fdsn webservice_fdsn
}
```

```
Run &Arr{
webservice_fdsn yes
}
```





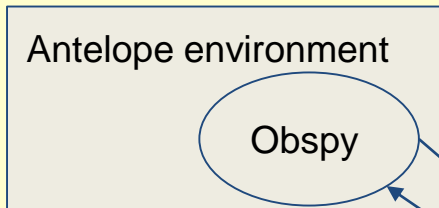
# FDSNWS Client Import in Antelope

- Obspy: Python framework for processing seismological data
- Antelope *Python* (*Python3!*) interpreter retrofitted with all modules necessary for *ObsPy*
- New contrib script *install\_obsipy(3)*
- *ObsPy* WS import capabilities
- Imports data via *ObsPy* but currently no provision for reformatting into *Datascope*





# Demo setup



Antelope FDSN Server

Station Information on selected data

- Antelope1/Agp1 (Load time: wait)
- Antelope1/Agp2 (Load time: wait)
- Antelope1/Agp3 (Load time: wait)

Data Select Information on selected data

- Antelope1/Agp1 (Load time: wait)
- Antelope1/Agp2 (Load time: wait)
- Antelope1/Agp3 (Load time: wait)

Event Information on selected data

- Antelope1/Agp1 (Load time: wait)
- Antelope1/Agp2 (Load time: wait)
- Antelope1/Agp3 (Load time: wait)

GNSS Antelope dev 2019-135 20:06

System is up

Load Average: 1min 1.49, 5min 1.63, 15min 2.02

Cpu Usage (20 cpus): 483 processes

Memory Usage: ram 131072 Mb, swap 65536 Mb

Disk Usage: root

Orb Ring Buffer Status: gsn pkts/s 17 connections

| Task                   | Pid          | cpu         | cpu          | rss         | rss         | To Orb   | To Orb | From Orb | From Orb | Latency        | Latency |
|------------------------|--------------|-------------|--------------|-------------|-------------|----------|--------|----------|----------|----------------|---------|
| rtexec                 | 86739        | 0.00        | 100.0        | 13.4        | 1500        |          |        |          |          |                |         |
| orbserver              | 86816        | 0.10        | 100.0        | 300.7       | 1500        |          |        |          |          |                |         |
| orbserver_mime         | 86829        | 0.00        | 100.0        | 305.2       | 1500        |          |        |          |          |                |         |
| orb2wf                 | 86848        | 0.40        | 100.0        | 33.6        | 1500        | 0.0 bps  | 20.00  | 9.1 Kbps | 20.00    | 22.293 seconds | 2000    |
| orb2dbt                | 86873        | 0.00        | 100.0        | 8.6         | 1500        | 0.0 bps  | 20.00  | 0.0 bps  | 20.00    | 1:05 minutes   | 2000    |
| orbdetect              | 87019        | 0.40        | 100.0        | 68.8        | 1500        | 0.0 bps  | 20.00  | 9.1 Kbps | 20.00    | 22.293 seconds | 2000    |
| orbassoc               | 87054        | 0.00        | 100.0        | 213.1       | 1500        | 0.0 bps  | 20.00  | 0.0 bps  | 20.00    | 1:05 minutes   | 2000    |
| magnitude              | 87096        | 0.00        | 100.0        | 69.0        | 1500        | 0.0 bps  | 20.00  | 0.0 bps  | 20.00    | 48.475 seconds | 2000    |
| GSNimport              | 87147        | 0.40        | 100.0        | 25.3        | 1500        | 8.6 Kbps | 20.00  | 0.0 bps  | 20.00    | 22.293 seconds | 2000    |
| USGSimport             | 87200        | 0.00        | 100.0        | 43.7        | 1500        | 0.0 bps  | 20.00  | 0.0 bps  | 20.00    | 2:45 minutes   | 2000    |
| rtcache                | 87262        | 0.10        | 100.0        | 62.5        | 1500        |          |        |          |          |                |         |
| rtwebservice           | 87387        | 0.00        | 100.0        | 33.8        | 1500        |          |        |          |          |                |         |
| <b>webservice_fdsn</b> | <b>88356</b> | <b>0.00</b> | <b>100.0</b> | <b>37.1</b> | <b>1500</b> |          |        |          |          |                |         |

Cron Job Status: patches, compress, cleanlogs, rtclean

Network Operation: processes, Orbstat, ORB\_Data, Dimon, DB\_data, Event\_Map, Grid\_Map, Stations\_Map

```
/opt/antelope/5.9$ curl http://127.0.0.1:5000/fdsnws/station/1/query?...
```



# Demo time!

Antelope FDSN Web-service reachable at: <http://<host-ip>:5000/fdsnws>

Please feel free to navigate to the above URL and check it out.

**Disclaimer: This is a work in progress, so not all parameters are supported.**



# Server example page

**Antelope FDSN Server** <sup>1.0</sup>  
[ Base URL: /fdsnws ]  
<http://127.0.0.1:5000/fdsnws/swagger.json>

**Station** Information on stationxml data ▼

- GET /station/1/application.wadl
- GET /station/1/query :return:
- GET /station/1/version

**Data Select** Information on datasetselect data ▼

- GET /datasetselect/1/application.wadl
- GET /datasetselect/1/query :return:
- POST /datasetselect/1/query :return:
- GET /datasetselect/1/version

**Event** Information on event data ▼

- GET /event/1/application.wadl
- GET /event/1/catalogs :return:
- GET /event/1/contributors :return:

<http://localhost:5000/fdsnws/>





Thank You!  
Questions?



# Feedback

Very early prototype: Questions? Concerns?  
Priorities?

- QuakeML import
- StationXML import
- Waveforms import
- QuakeML export
- StationXML export
- Waveforms export



# References

- FDSN Web-service specification: <http://www.fdsn.org/webservices/FDSN-WS-Specifications-1.2.pdf>
- OpenAPI 2.0: <https://swagger.io/specification/>
- REST: [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)