

# Using STOQS\* to manage, visualize, and understand AUV, glider, and mooring data

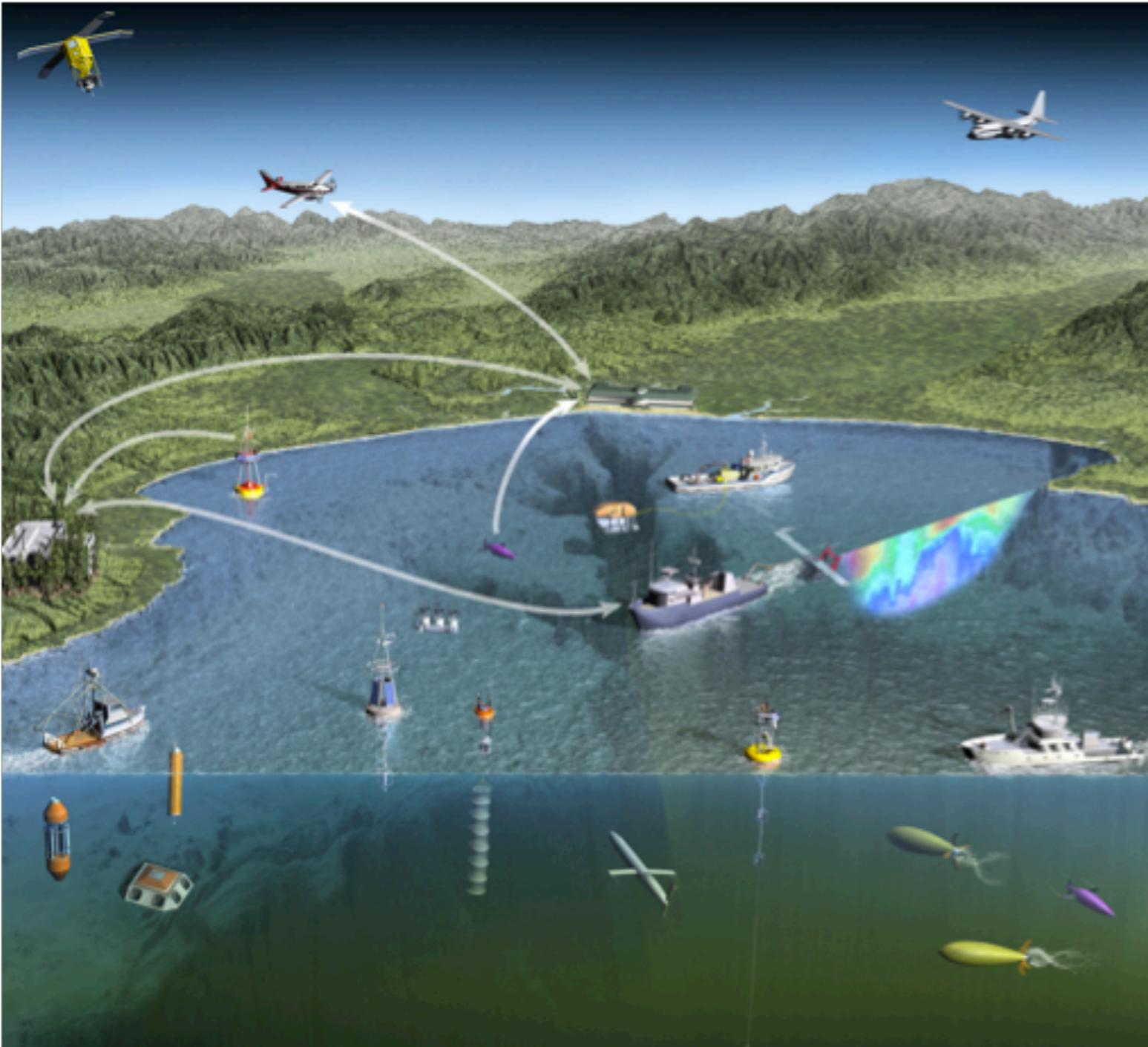
\*Spatial Temporal Oceanographic Query System

8 October 2014  
IEEE/OES AUV 2014  
Oxford, Mississippi

Mike McCann  
Monterey Bay Aquarium Research Institute



# Multi-platform Campaigns



10/8/2014 11:34 am



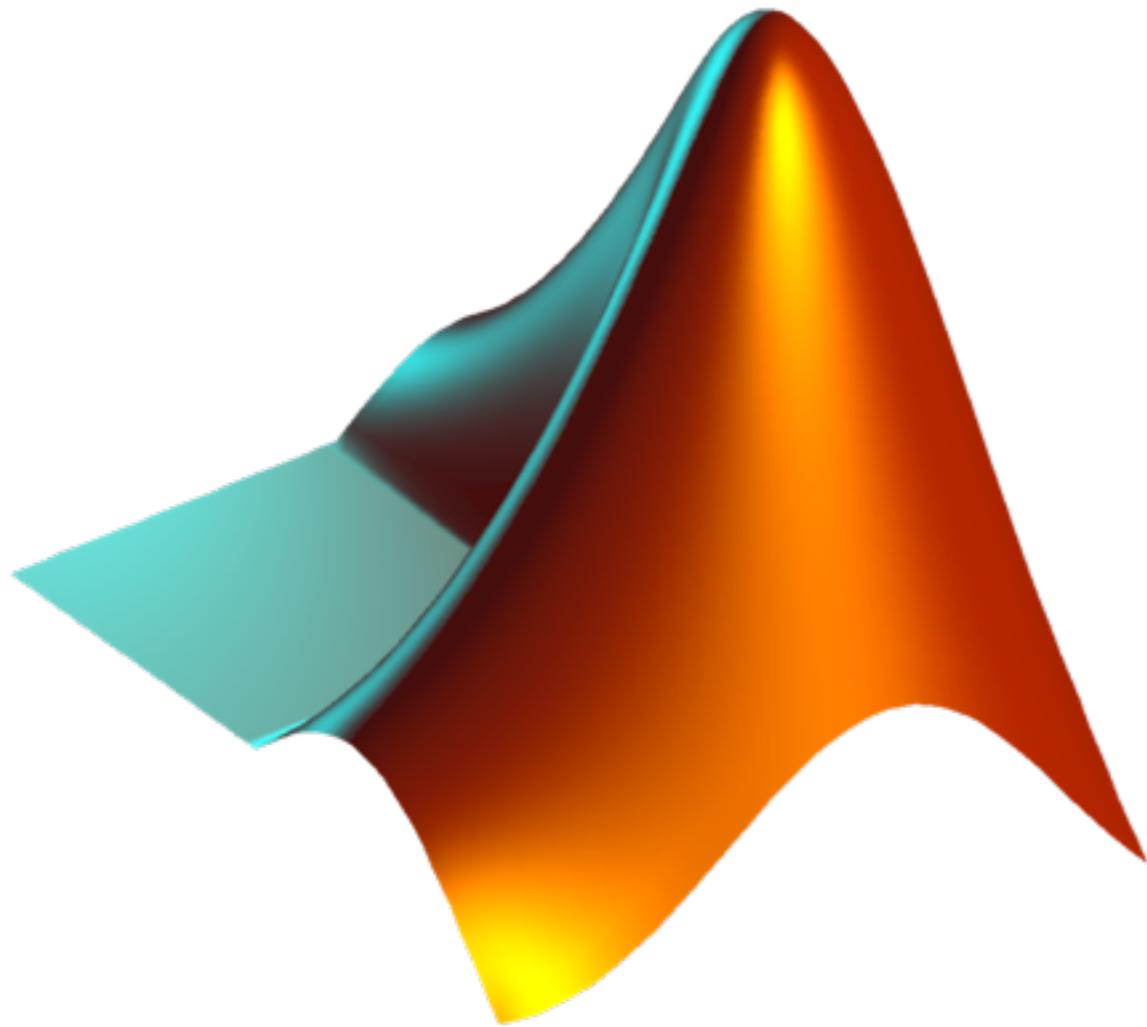
dora ARION thys  
makai\_ac Gordon  
makai 367766000  
M46044 366974890  
oal

Data CSUMB SFML, CA OPC  
Image Landsat  
Data MBARI  
Data SIO, NOAA, U.S. Navy, NGA, GEBCO

Imagery Date: 4/9/2013 36°46'18.77" N 122°13'41.76" W elev -665 m eye alt 90.62 km



# Problem



# Web Database App

- Use standards and community conventions
- Use free and open source software
- Make available for anyone to use



# STOQS

← → C <https://code.google.com/p/stoqs/>

 stoqs  
Spatial Temporal Oceanographic Query System

[Project Home](#) [Downloads](#) [Wiki](#) [Issues](#) [Source](#) [Administer](#)

[Summary](#) [People](#)

**Project Information**

 Starred by 6 users [Project feeds](#)

**Code license** [GNU GPL v3](#)

**Labels** [Django](#), [Python](#), [Research](#), [KML](#), [GIS](#), [OPeNDAP](#), [NetCDF](#), [AJAX](#), [Mapping](#), [Mapserver](#), [OpenLayers](#), [Oceanography](#), [data](#), [X3D](#)

 **Members** [MBARImike](#) 8 committers

**About STOQS**

STOQS (Spatial Temporal Oceanographic Query System) is a geospatial database platform designed for providing efficient access to *in situ* oceanographic measurement data across any dimension. Where "dimension" is considered in the broadest sense: any spatial dimension, time, parameter, platform, or any measured parameter data value.

**Use Case**

**Precondition:**  
Observational data from a variety of platforms (Autonomous Underwater Vehicles, gliders, moorings, drifters, shipboard systems) are individually available from OPeNDAP servers.

**Goal:**  
Retrieve data by parameter name across all platforms in a specific area over a specific span of time



# CF-NetCDF 1.6 DSG

## Trajectory

```
Dataset {
    Float64 time[time = 50355];
    Float64 latitude[time = 50355];
    Float64 longitude[time = 50355];
    Float64 depth[time = 50355];
    Float64 temperature[time = 50355];
    Float64 oxygen[time = 50355];
    Float64 nitrate[time = 50355];
    Float64 bbp420[time = 50355];
    Float64 bbp700[time = 50355];
    Float64 f1700_uncorr[time = 50355];
    Float64 salinity[time = 50355];
    Float64 biolume[time = 50355];
} auv/dorado/2010/netcdf/Dorado389_2010_300_00_300_00_decim.nc;
```

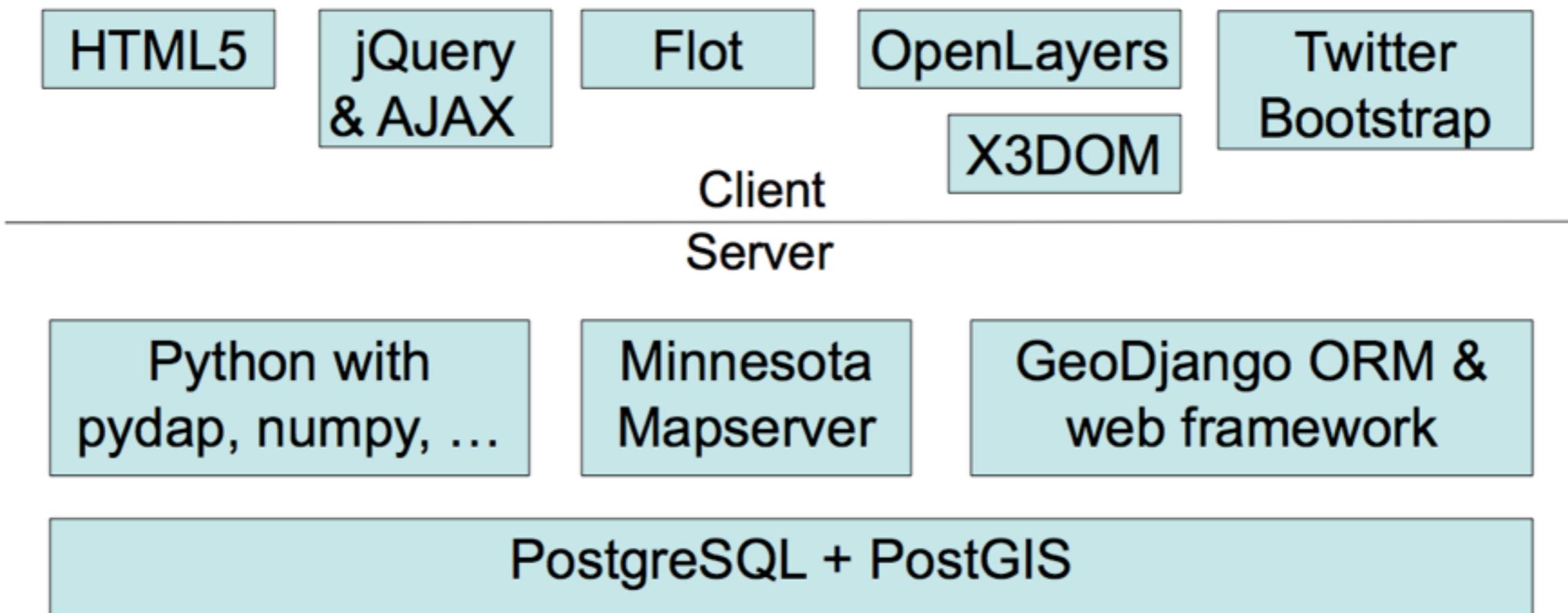


# Workflow

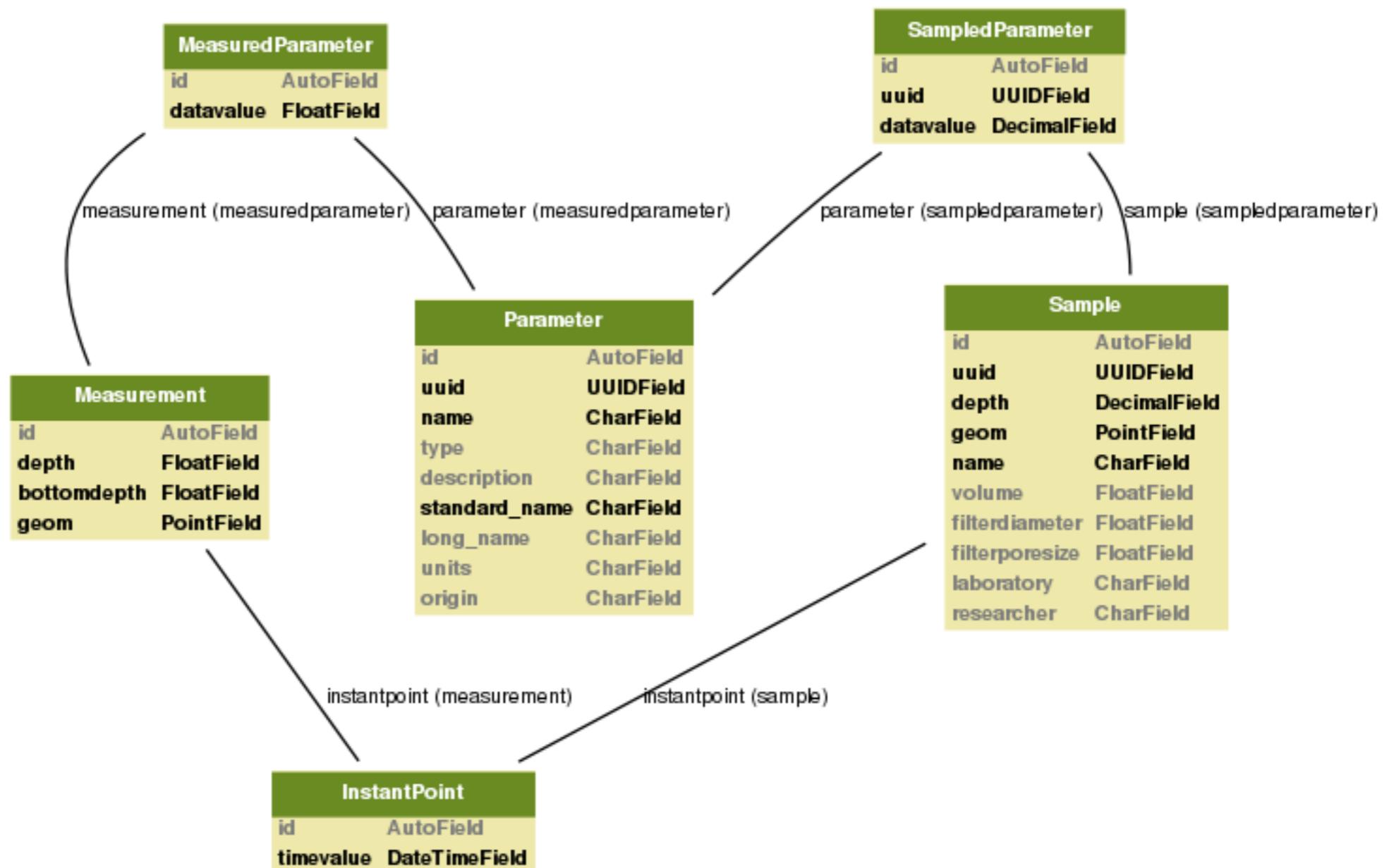
1. Install STOQS from [stoqs.googlecode.com](http://stoqs.googlecode.com)
2. Conduct missions that collect data
3. Create CF-NetCDF 1.6 files of the data
4. Construct STOQS load script
5. Create PostgreSQL database and run script
6. Explore, visualize, and understand data

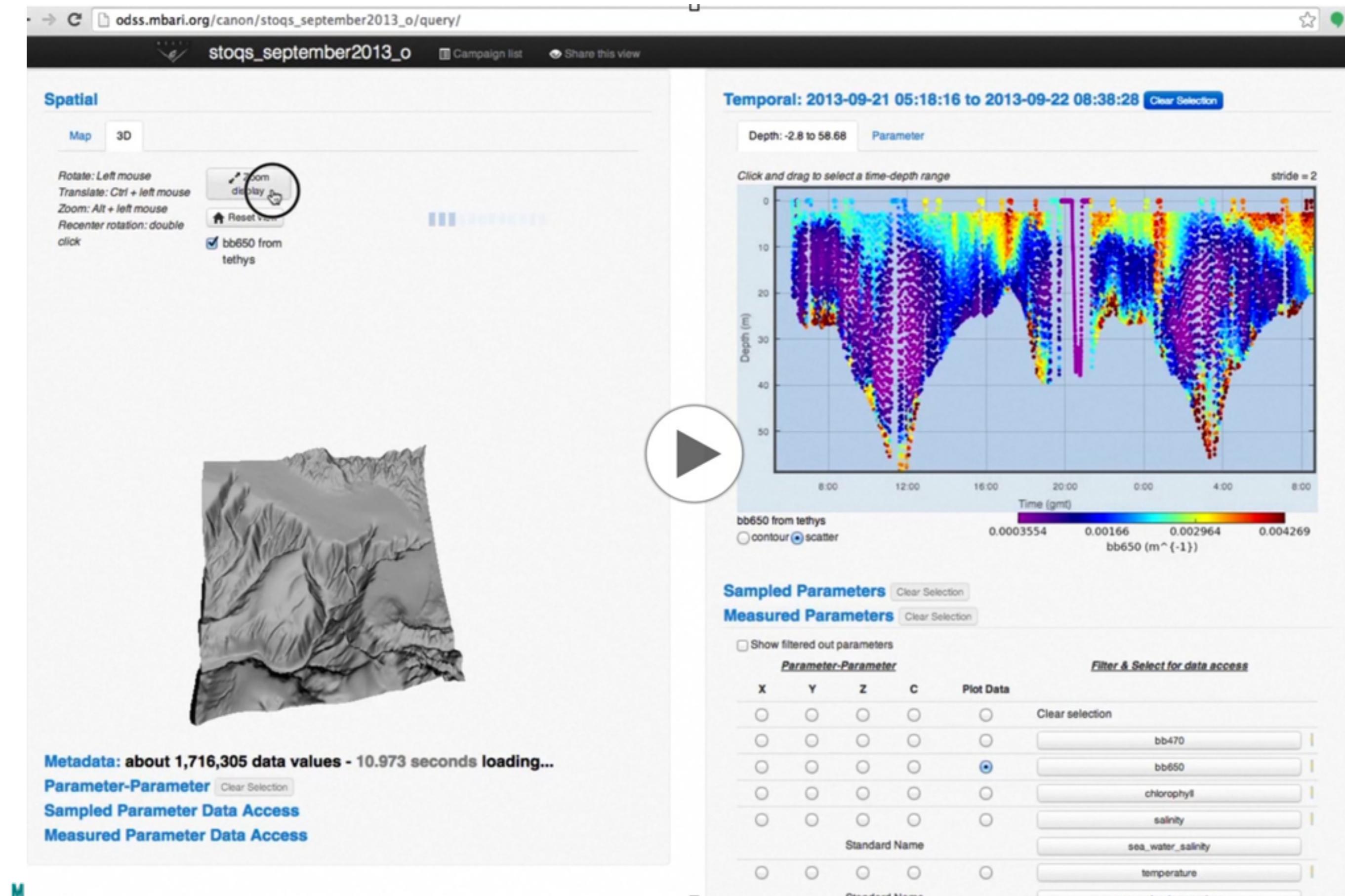


# STOQS Architecture



# Relational Database



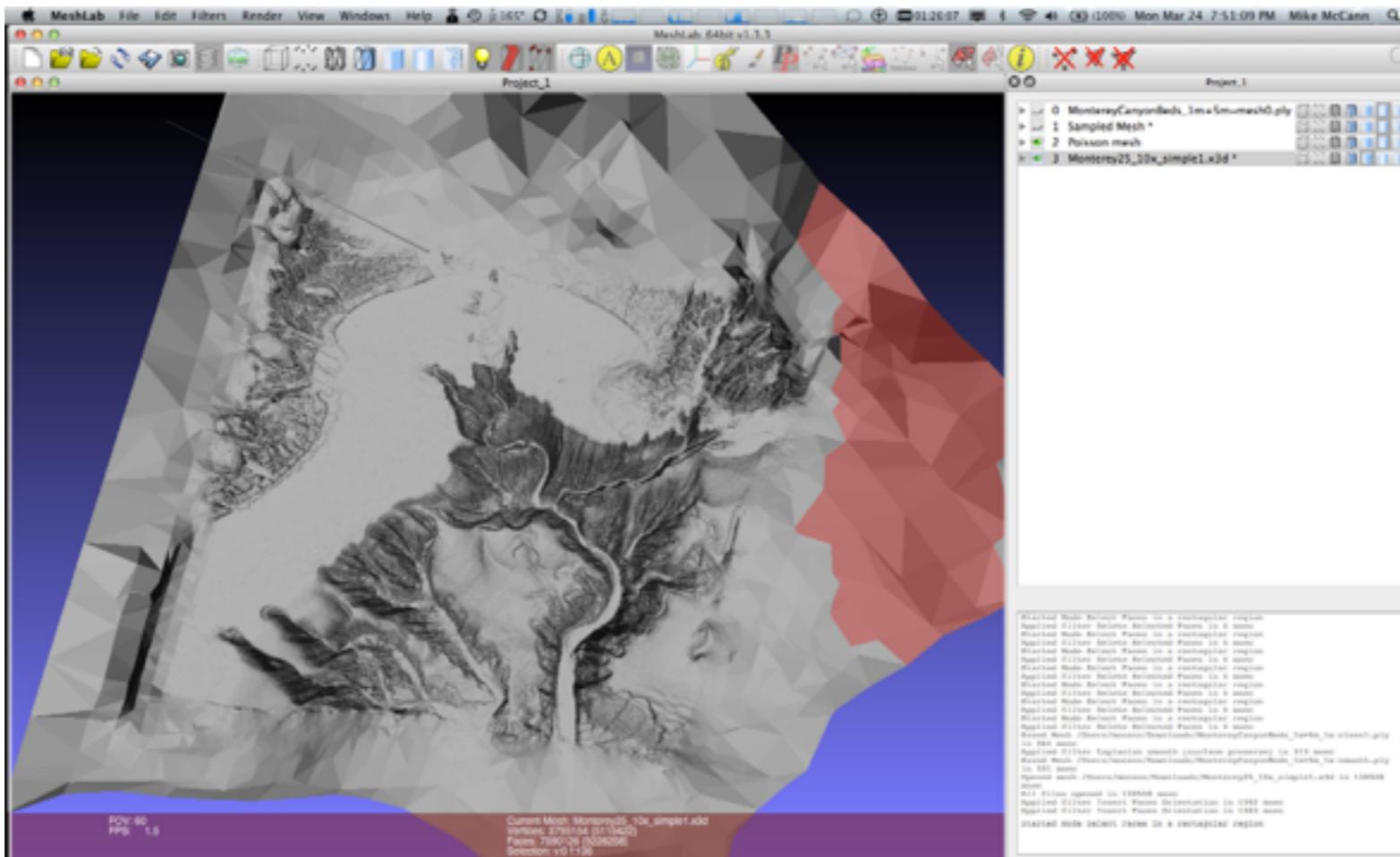


# Terrain Generation (I)

- Convert GMT file to point cloud in GCC

```
grd2xyz Monterey25.grd --D_FORMAT=%f | sed \
' /NaN/d' | awk '{print $1, $2, 10 * $3}' | \
mapproject -E > Monterey25_10x.asc
```

- Process and edit in Meshlab



# Terrain Generation (2)

- InstantReality aopt execution (on Mac) to convert mesh to PopGeometry

```
aopt -i Monterey25_10x-clean.ply -F Scene \
    -b Monterey25_10x-opt.x3db
```

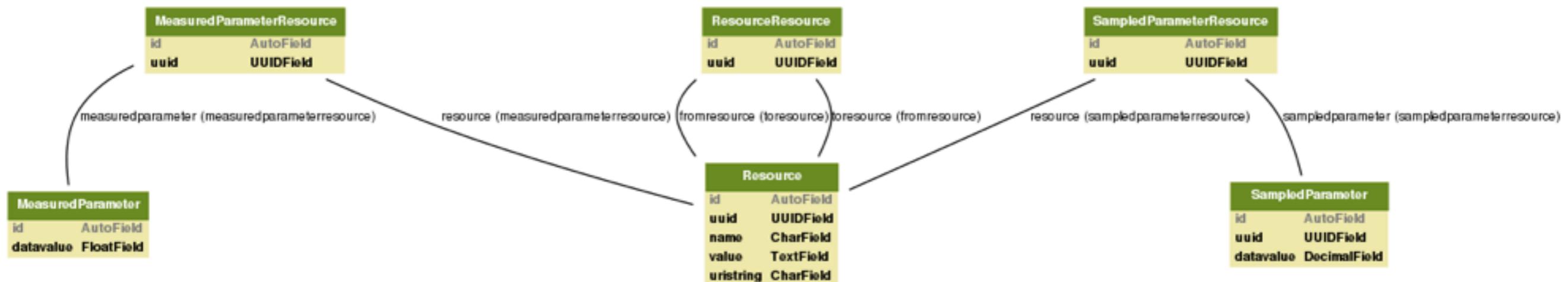
```
aopt -i Monterey25_10x-opt.x3db -f \
    PrimitiveSet:creaseAngle:4 -V -K \
    "binGeo/:ib" -N Monterey25_10x.html
```

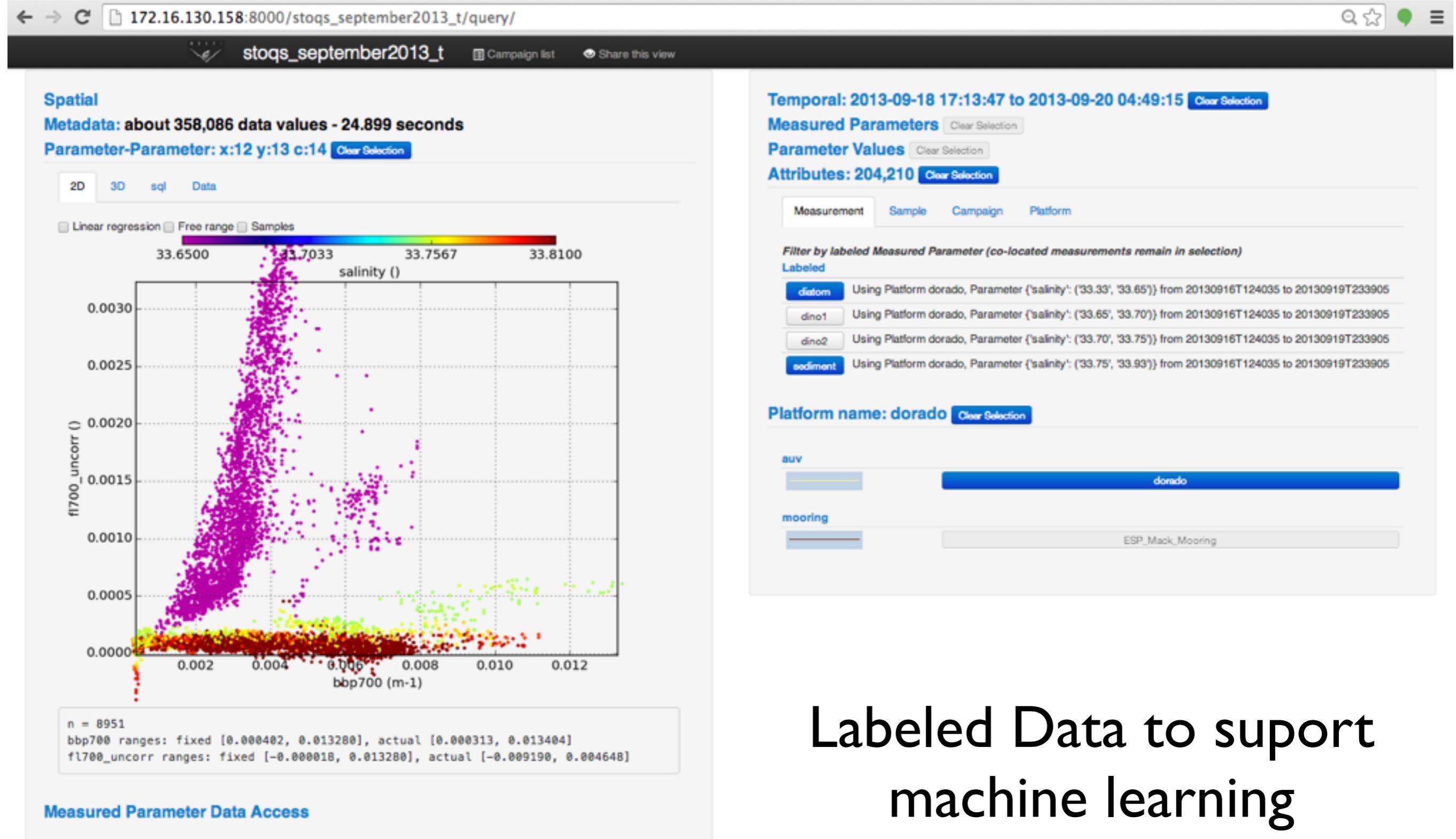


# Live Demo



# Tables to support machine learning





# Labeled Data to support machine learning

