

# Wilber 3: A Python-Django Web Application For Acquiring Large-scale Event-oriented Seismic Data

Robert Newman (rnewman@iris.washington.edu), **Adam Clark (adam@iris.washington.edu)**, Chad Trabant, Rich Karstens, Alex Hutko, Rob Casey, Tim Ahern  
IRIS Data Management Center, Seattle, WA 98105

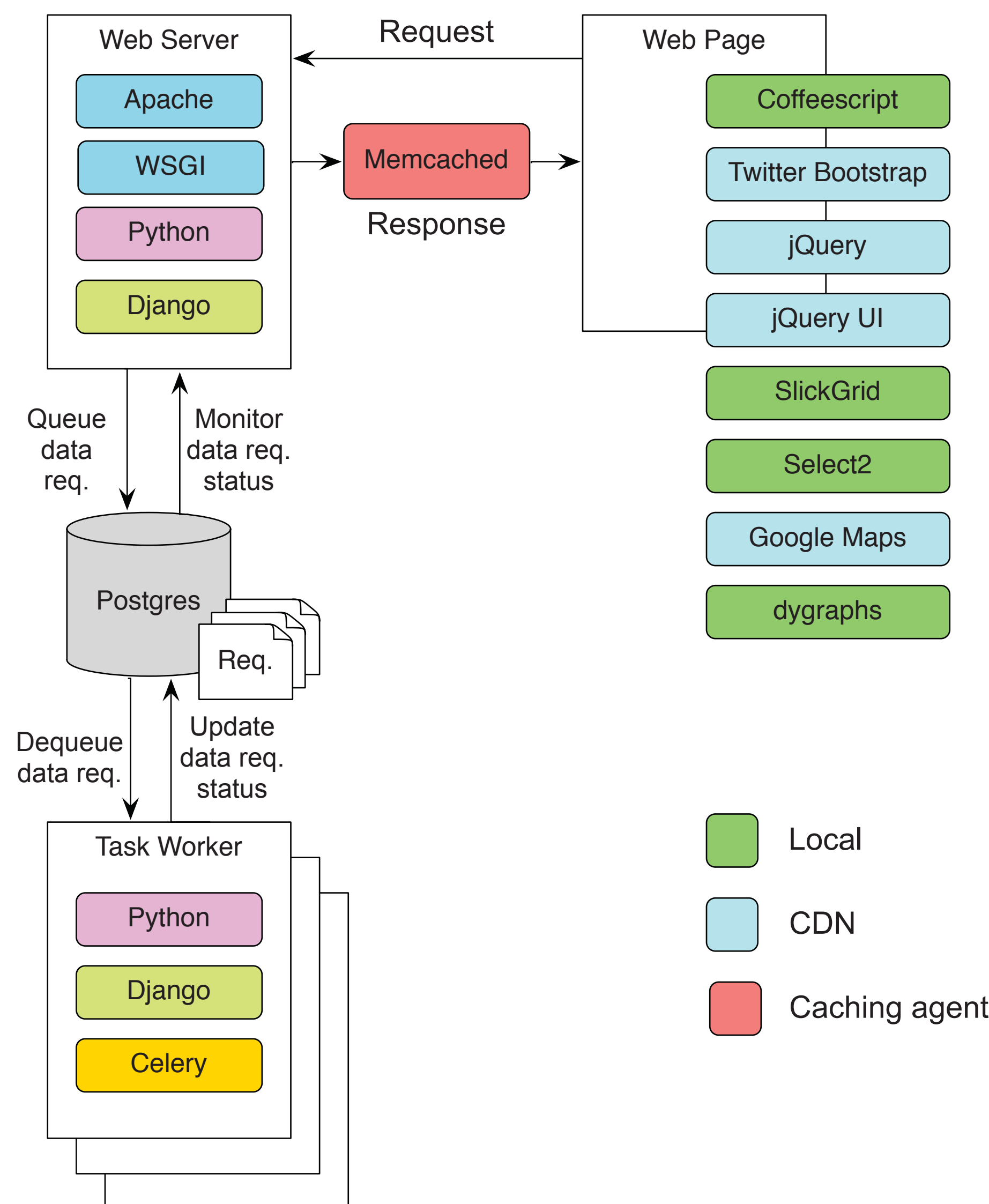
## 1. Abstract

Since 2001, the IRIS Data Management Center (DMC) WILBER II system has provided a convenient web-based interface for locating seismic data related to a particular event, and requesting a subset of that data for download. Since its launch, both the scale of available data and the technology of web-based applications have developed significantly. Wilber 3 is a ground-up redesign that leverages a number of public and open-source projects to provide an event-oriented data request interface with a high level of interactivity and scalability for multiple data types.

Wilber 3 uses the IRIS/Federation of Digital Seismic Networks (FDSN) web services for event data, metadata, and time-series data. Combining a carefully optimized Google Map with the highly scalable SlickGrid data API, the Wilber 3 client-side interface can load tens of thousands of events or networks/stations in a single request, and provide instantly responsive browsing, sorting, and filtering of event and meta data in the web browser, without further reliance on the data service.

The server-side of Wilber 3 is a Python-Django application, one of over a dozen developed in the last year at IRIS, whose common framework, components, and administrative overhead represent a massive savings in developer resources. Requests for assembled datasets, which may include thousands of data channels and gigabytes of data, are queued and executed using the Celery distributed Python task scheduler, giving Wilber 3 the ability to operate in parallel across a large number of nodes.

## 2. Schematic data flow from request

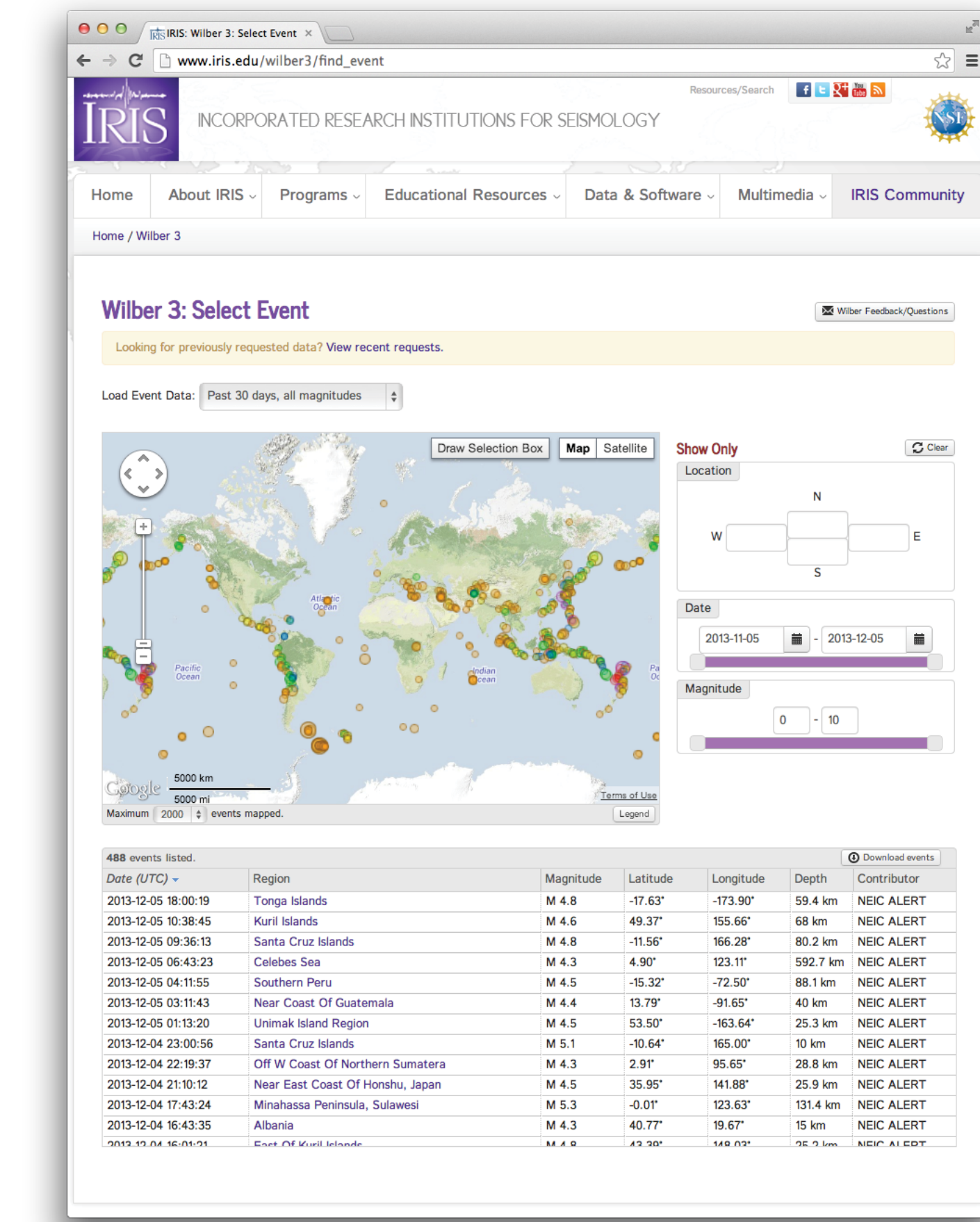


## 3. Features

- Real-time processing using AJAX & FDSN/IRIS webservices.
- Processing methodology based on data request type (eg. SEED vs. others).
- Queue Prioritization: **Celery** is a distributed task queue manager and is focused on real-time operation. Priority is configurable. Currently based on request size (number of stations & channels).
- Use of Content Delivery Networks (CDN) for Javascript libraries improves performance.
- Use of Memcached (a memory object mapping system that uses a key-value store) for optimizing dynamic delivery.
- User preferences stored (preferred networks, user profile).

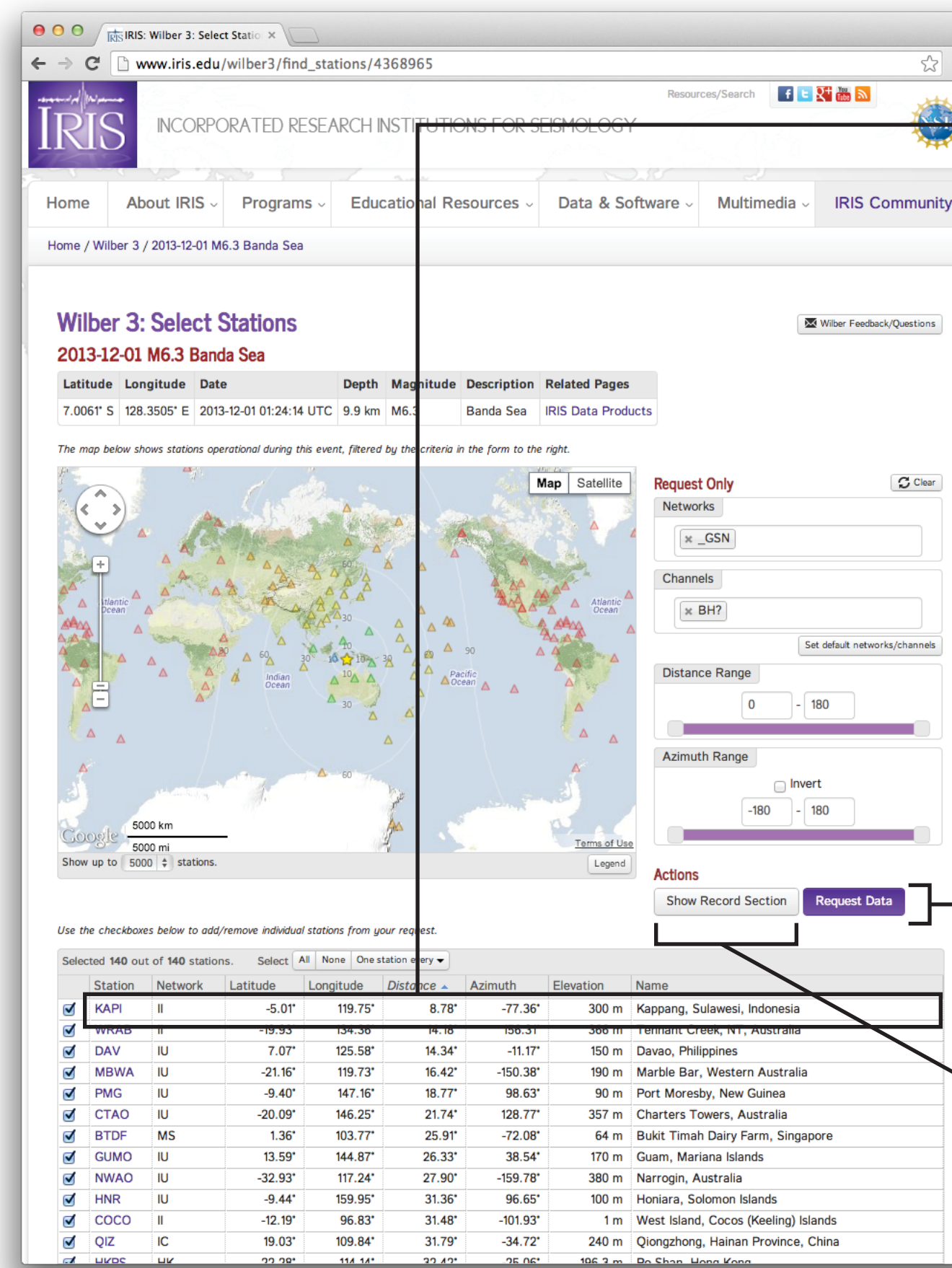
## 4. Screenshots of UI Panels, Data Preview, Data Request & Download

### Events

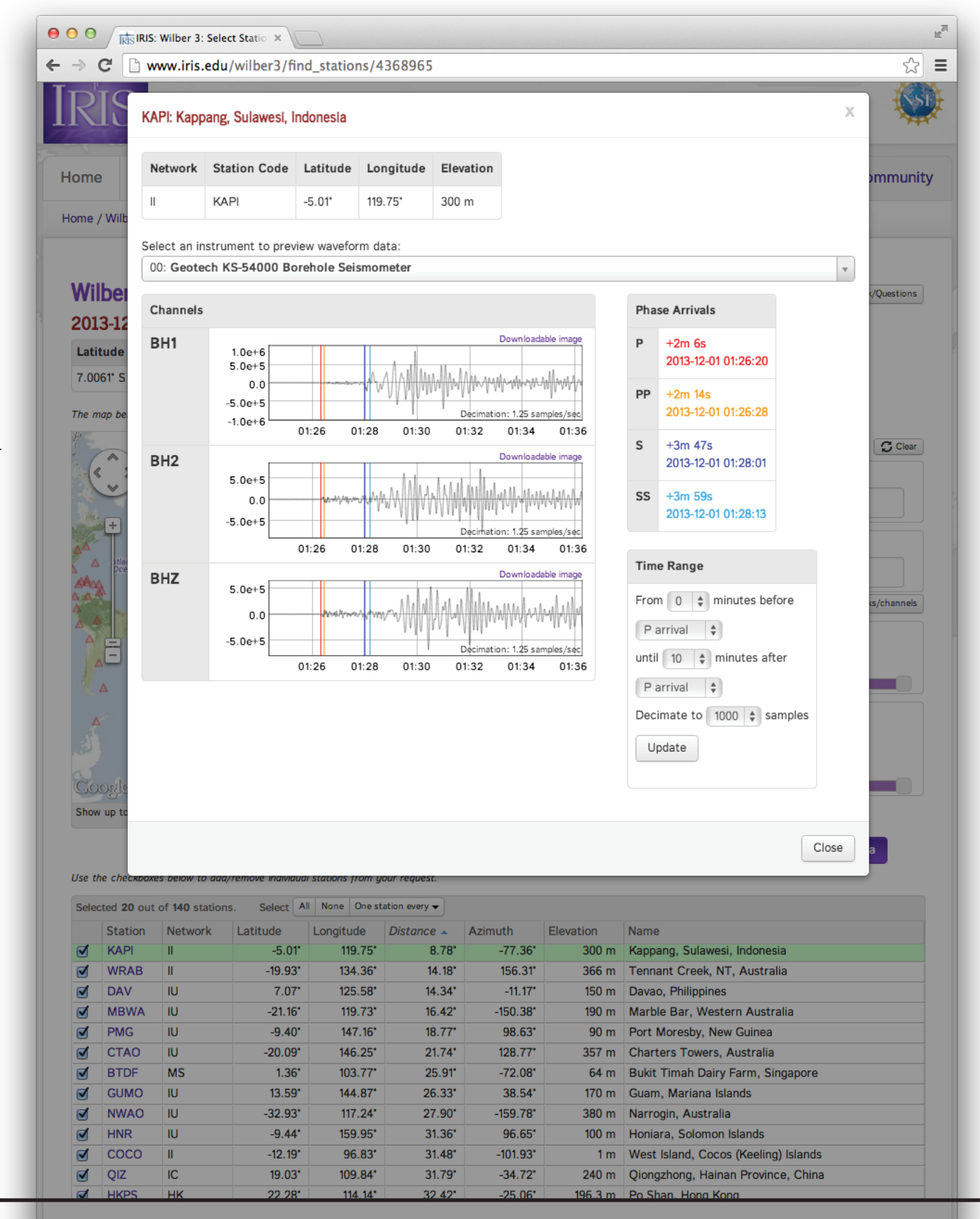


Select Event

### Stations



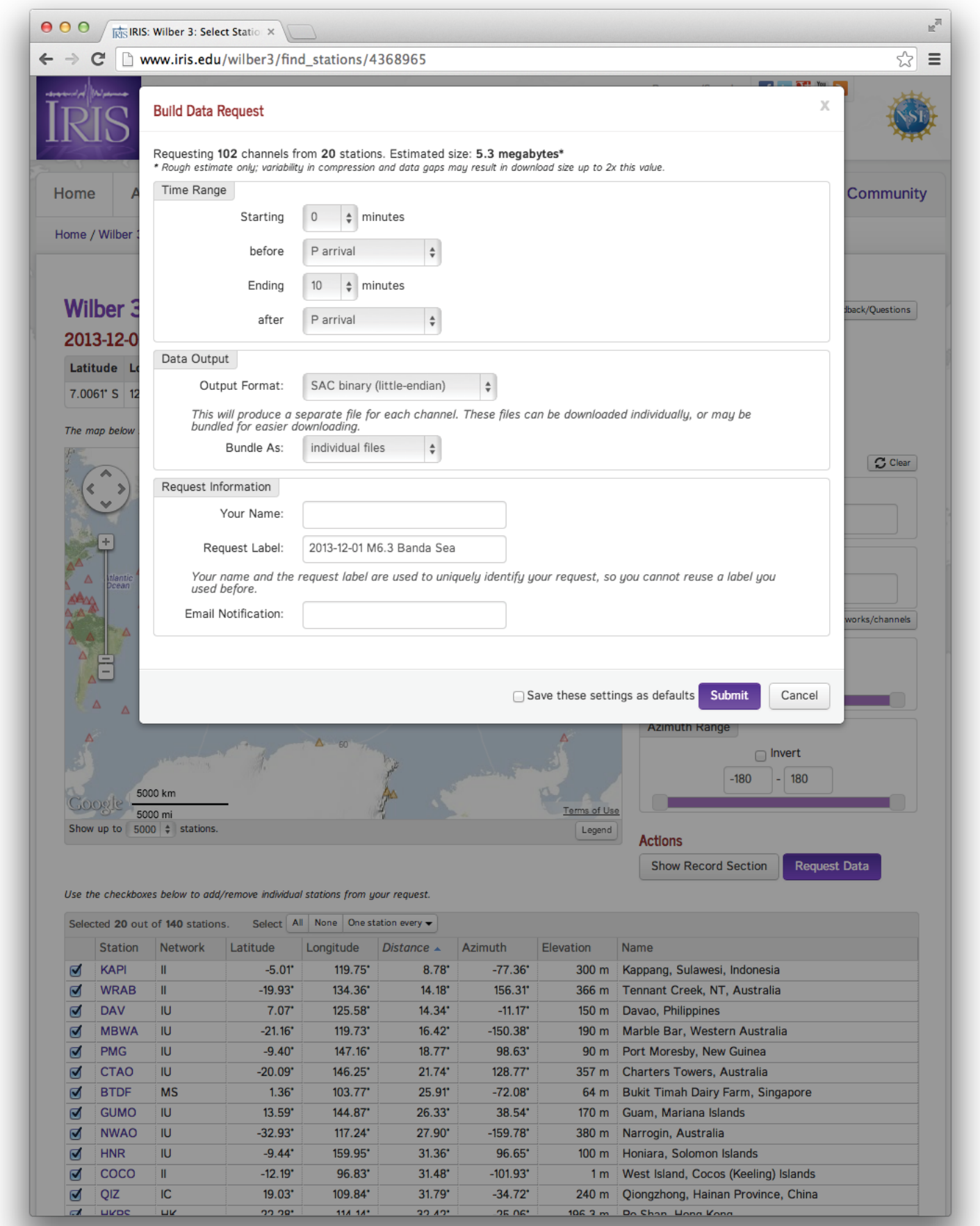
Per Station Record Section



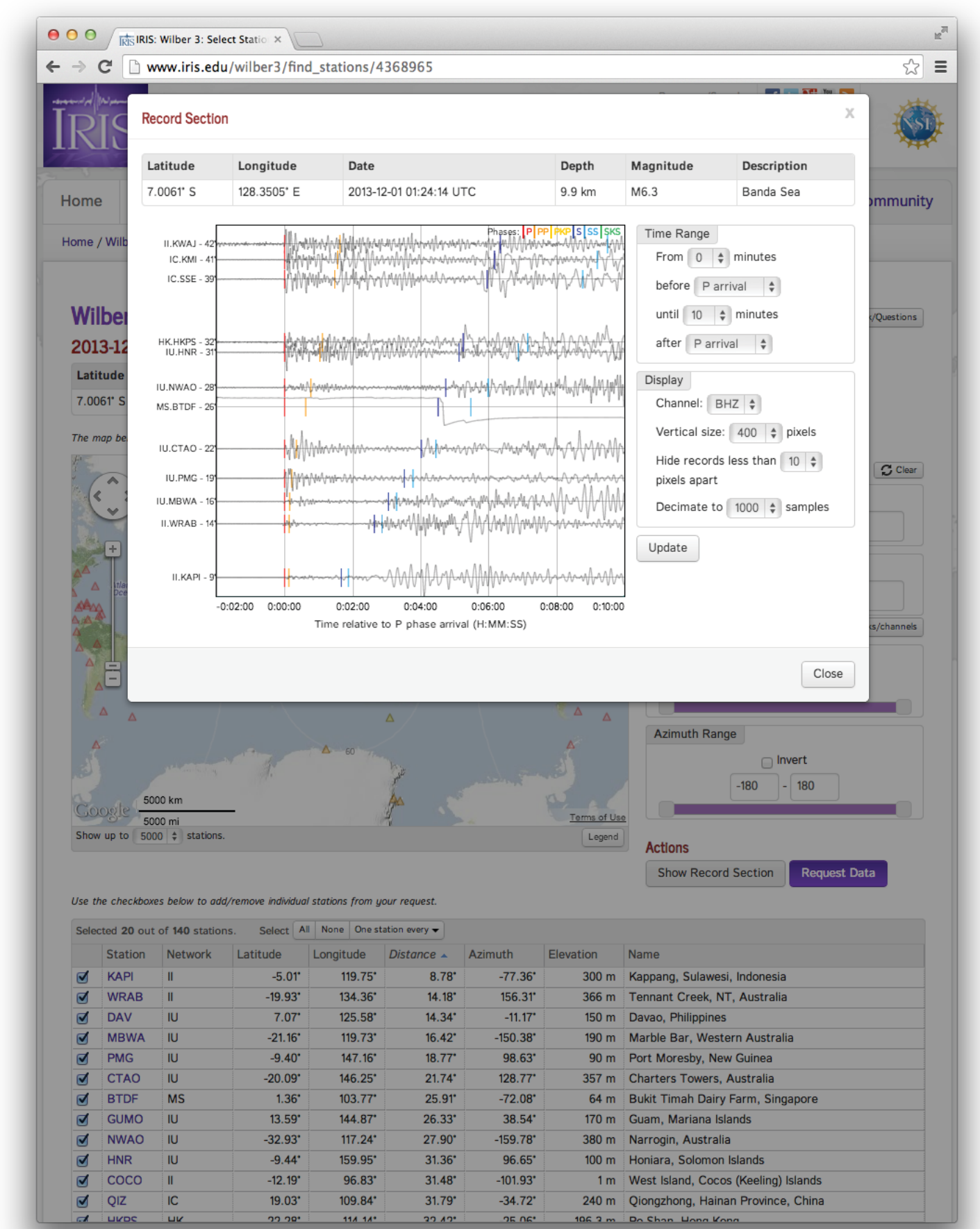
### Features:

- Dynamic plots for different instruments
- Predicted arrivals from IRISWS traveltime
- Record sections plots from IRISWS timeseries
- Plotting by dygraph Javascript library

## Build Data Request



## Data Preview



Vertical Record Sections For Selected Stations

### Features:

- Predicted arrivals from IRISWS traveltime
- Record sections plots from IRISWS timeseries
- Plotting by dygraph Javascript library

## 5. Data Downloaded Statistics: WILBERII vs. Wilber3 Comparison

The Web Interface to Lookup Big Events for Retrieval (WILBER) is a web-based tool for previewing data before requesting direct data download. Wilber3 is the replacement for WILBERII, a Perl CGI interface written in 2000 to view pre-created maps, listings, and seismogram plots for large events. WILBERII was phased out, and Wilber3 phased in, over 2013 as shown by the data shipment plots below. In July 2013, over 0.5 TB of data were shipped by Wilber3.

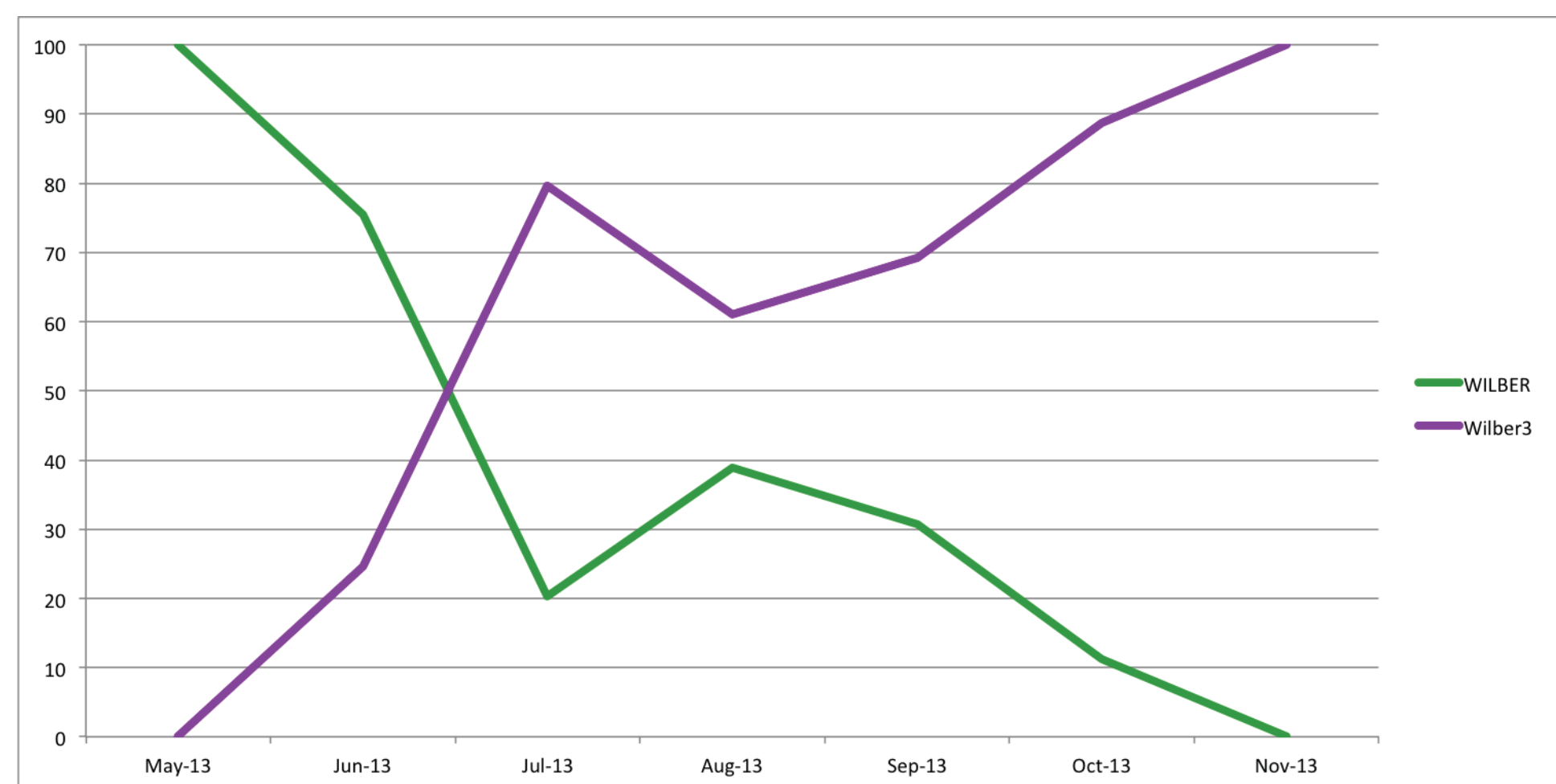


Figure 1: Data shipped (as a percentage of all WILBER shipments) comparison between WILBERII and Wilber3 from May 2013 (when Wilber3 came online) and November 2013 (when WILBERII was decommissioned).

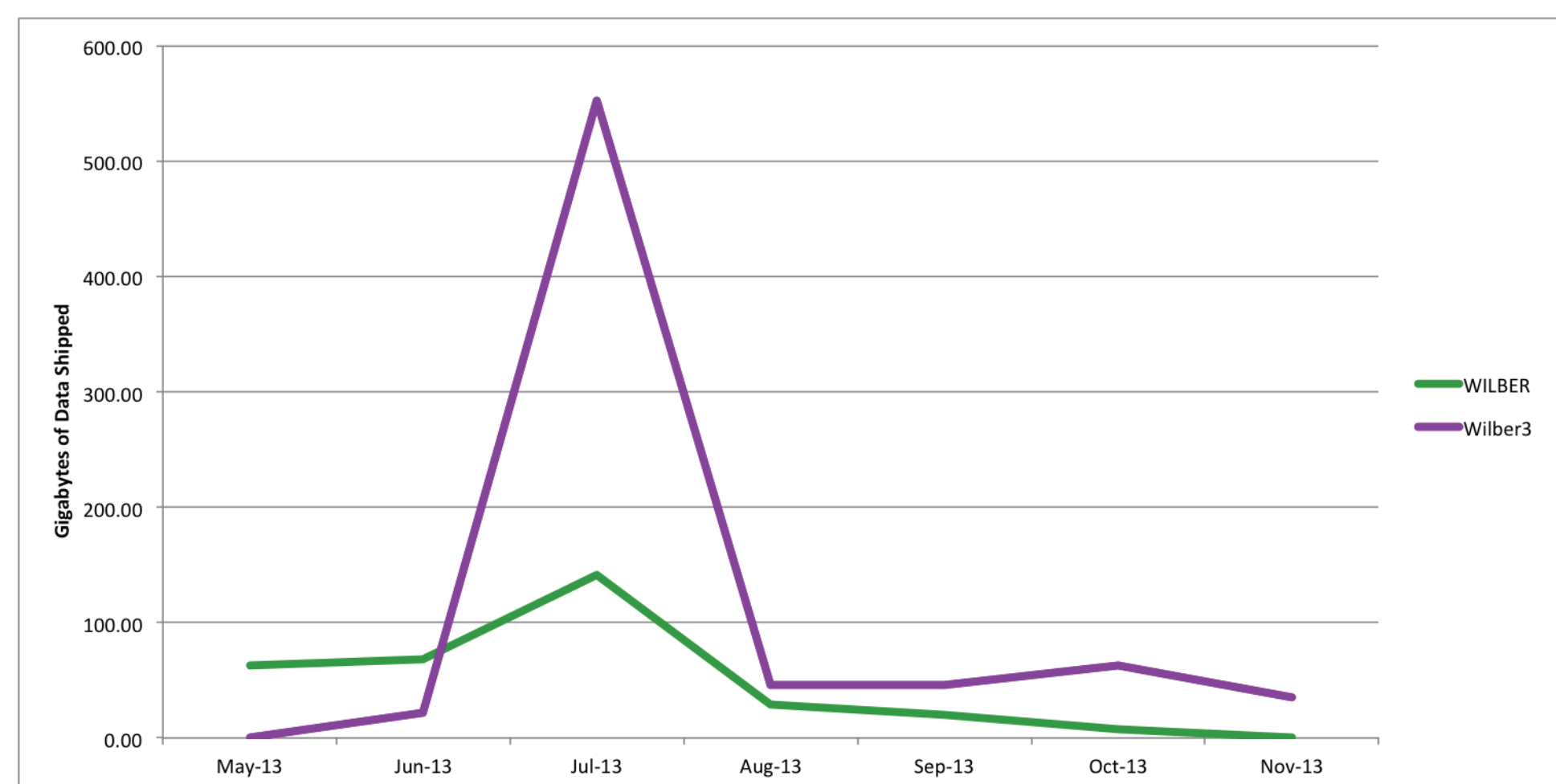


Figure 2: Gigabytes of data shipped comparison between WILBERII and Wilber3 from May 2013 (when Wilber3 came online) and November 2013 (when WILBERII was decommissioned).

## 6. Pros & Cons

Using open-source software has advantages and disadvantages. In the context of this web-application, we have the following bullet points for discussion:

### PROS

- Free
- Works out of the box (most of the time)
- Good support (most of the time)
- Source-code transparency makes debugging & patching (pull requests) relatively simple (no black boxes)

### CONS

- Packages do not necessarily play well with others: name-space clashes, DOM-manipulation clashes, out-of-sync package updates
- High-flux changes: Not necessarily backwards compatible
- You don't always get exactly what you want = hacking

## 7. Potential future optimizations

- Static assets pipeline (icon sprites, minimized/combined JS & CSS)
- Static assets via alternative webserver (eg. Nginx)
- Better plotting engine

## 8. Feature proposals

- Working with data and requesting data for multiple events
- Support restricted (pre publication) data
- Support federated FDSN data center services
- Modularize components for use in other packages/projects

## 9. References

**Wilber3:** <http://www.iris.edu/wilber3>  
**FDSN & IRIS Web Services:** <http://service.iris.edu>  
**Django:** <https://www.djangoproject.com/>  
**Celery Distributed Task Queue:** <http://www.celeryproject.org/>  
**Memcached:** <http://memcached.org/>