CGSN Dashboard
(OMS++ and ERDDAP)
Updated CGSN/EA Marine Operator Interface

Stephanie M. Petillo, PhD
Christopher Wingard, MS
What is the CGSN Dashboard?

• Next iteration on the current OMS mooring user interface
  • Used to monitor the Pioneer, Endurance, and Global Surface & Profiler moorings

• CGSN Dashboard (OMS++):
  • Improve and expand upon OMS functionality
  • Add more features to reduce the operators' status reporting and monitoring loads
  • Include Subsurface mooring, Glider/AUV, & Coastal Surface Piercing Profiler status displays (future)
Capabilities

- Automated parsing & processing of raw mooring data, including
  - Automated calibration lookup
  - Back-processing capability
- Parsed & processed data available to mooring operators for monitoring & analysis via:
  - Direct access to files via web page
  - ERDDAP
  - Plot creation and display (OMS++)
  - Calculation and display of L3 variables (OMS++)
- Configurable automated alerts & alarms and notification system
- Alert trigger, plot, & L3 variable cloning
- Support for multiple deployments of a mooring
- Uses Yaml files for simplified configuration of mooring deployments and specific asset metadata
- System overview & status pages
Capabilities (cont.)

• Access to external tools
  • Redmine
  • Roundabout Database (asset tracking system, in development)

• Links to Glider & AUV monitoring & status (in development)

• Uses up-to-date software and operating systems

• Uses open source tools to provide:
  • Visibility into data provenance
  • Ability to reproduce system

• Future: Offline laptop deployment capability for shipboard use

• Future: Status & monitoring tools for Subsurface Moorings & Surface Piercing Profilers
CGSN-PARSERS: Logs → JSON

- Python modules & shell scripts
- Parse the raw log files to create commonly formatted, human & machine readable JSON data files
- Puts raw data into a common format for further work
  - No unit conversions, no calculation of new, derived variables, no QA/QC, etc.
- Parsers exist for all Surface, Profiler, & CSPP Mooring instruments & engineering sensors
- Code publically available and hosted on BitBucket (conda-forge): (https://bitbucket.org/ooicgsn/cgsn-parsers)
CGSN-PROCESSING: JSON ➔ NetCDF

• Python modules and shell scripts
• Converts JSON formatted raw data files created via cgns-parsers into NetCDF4 datasets, served via ERDDAP
• Utilizes pre-existing ion-functions code forked to more simplified, generic code base
  • Convert values (e.g. counts to mg/L)
  • Derive new values (e.g. practical salinity, pH, OPTAA spectra)
  • [https://bitbucket.org/ooicgsn/pyseas](https://bitbucket.org/ooicgsn/pyseas)
• Utilizes dictionaries of NetCDF attributes to set common metadata for data set variables
  • Tedious to create, but one-and-done operation under version control
• Processors exist for all Surface, Profiler, & CSPP Mooring instruments & engineering sensors, except:
  • VEL3D (need to correct instrument clock)
  • Inductive: CTDBP & CTDMO (in progress), PHSEN & PCO2W (may only need to tweak non-inductive processor)
• Code publically available and hosted on BitBucket: [https://bitbucket.org/ooicgsn/cgsn-processing](https://bitbucket.org/ooicgsn/cgsn-processing)
PYSEAS: Data Product Algorithms

• Fork of current ion-functions code used in OOI production system
  • Converted to python 3.
  • Removing CI customizations (e.g. wrapper functions), creating more generic code that can serve as python toolbox for processing data from oceanographic sensors.
  • Removed use of OS dependent code and adopted packages that are OS independent (GSW, IGRF12).
  • Removed older QA/QC code in favor of adopting IOOS QARTOD code.
• Code publically available and hosted on BitBucket:
  ( https://bitbucket.org/ooicgsn/pyseas)
System Architecture Diagram
CGSN Dashboard Demo
Questions?