

Software Deployment in the Field: Technical Debt and Data-Ops

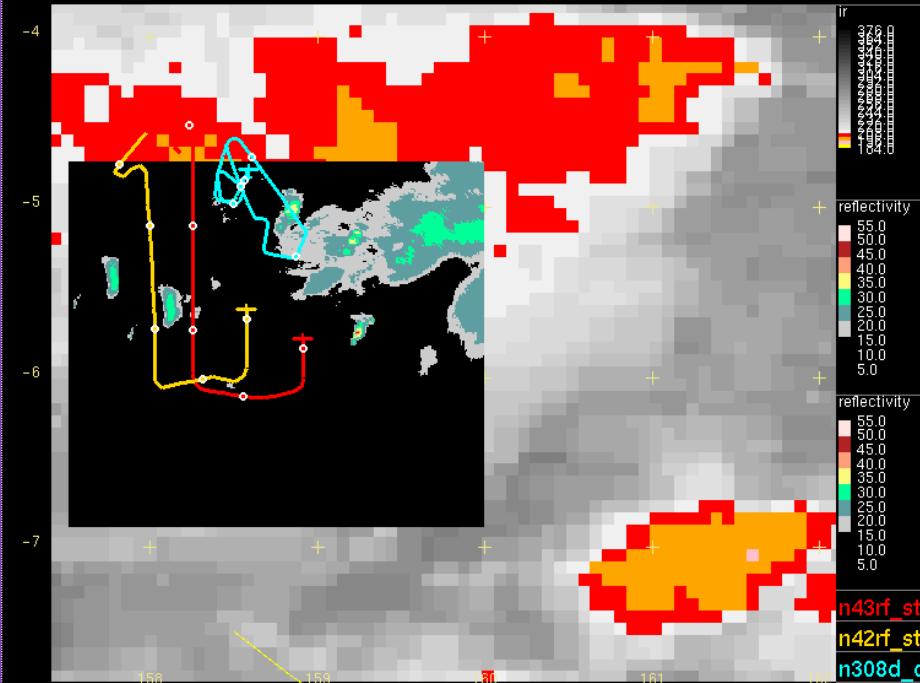
SEA Software Engineering Conference
April 14, 2015

Gary Granger
Earth Observing Laboratory
NCAR

Stories from the Field (and a big mistake from my past)

TOGA-COARE in Honiara, Solomon Islands

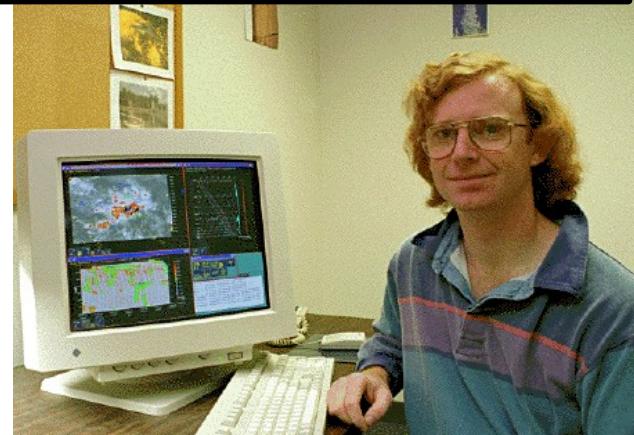
10-feb-1993,20:31:00 ir plot.radar reflectivity plot.radar reflectivity
plot.n43rf_st track. n42rf_st track. n308d_dap track.



Alt: 0m

COMPONENT	PLATFORM	FIELD	TIME
c_gms	gms-big	ir	10-Feb-93,19:33:00
c_belly	n42rf_if	reflectivity	10-Feb-93,20:24:5
c_belly_0	n43rf_if	reflectivity	10-Feb-93,20:26:3
c_slow_0	n43rf_st		10-Feb-93,20:31:00
c_slow	n42rf_st		10-Feb-93,20:31:00
c_dap	n308d_dap		10-Feb-93,20:31:00

Mesoscale Group
Department of Atmospheric Sciences
University of Washington
with NCAR, NOAA, and NASA



Mobile Sounding Launch at VORTEX-II



R/V Knorr in CLIMODE



Coding Can Happen Anywhere

Sea Containers



Rental Trailers

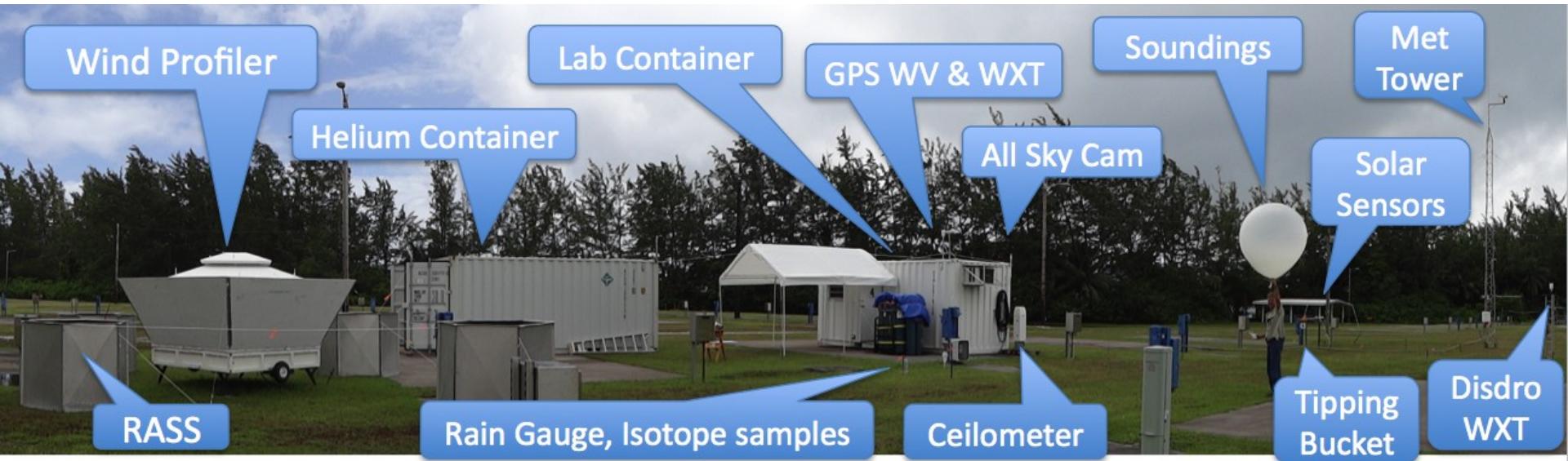


Ship Labs

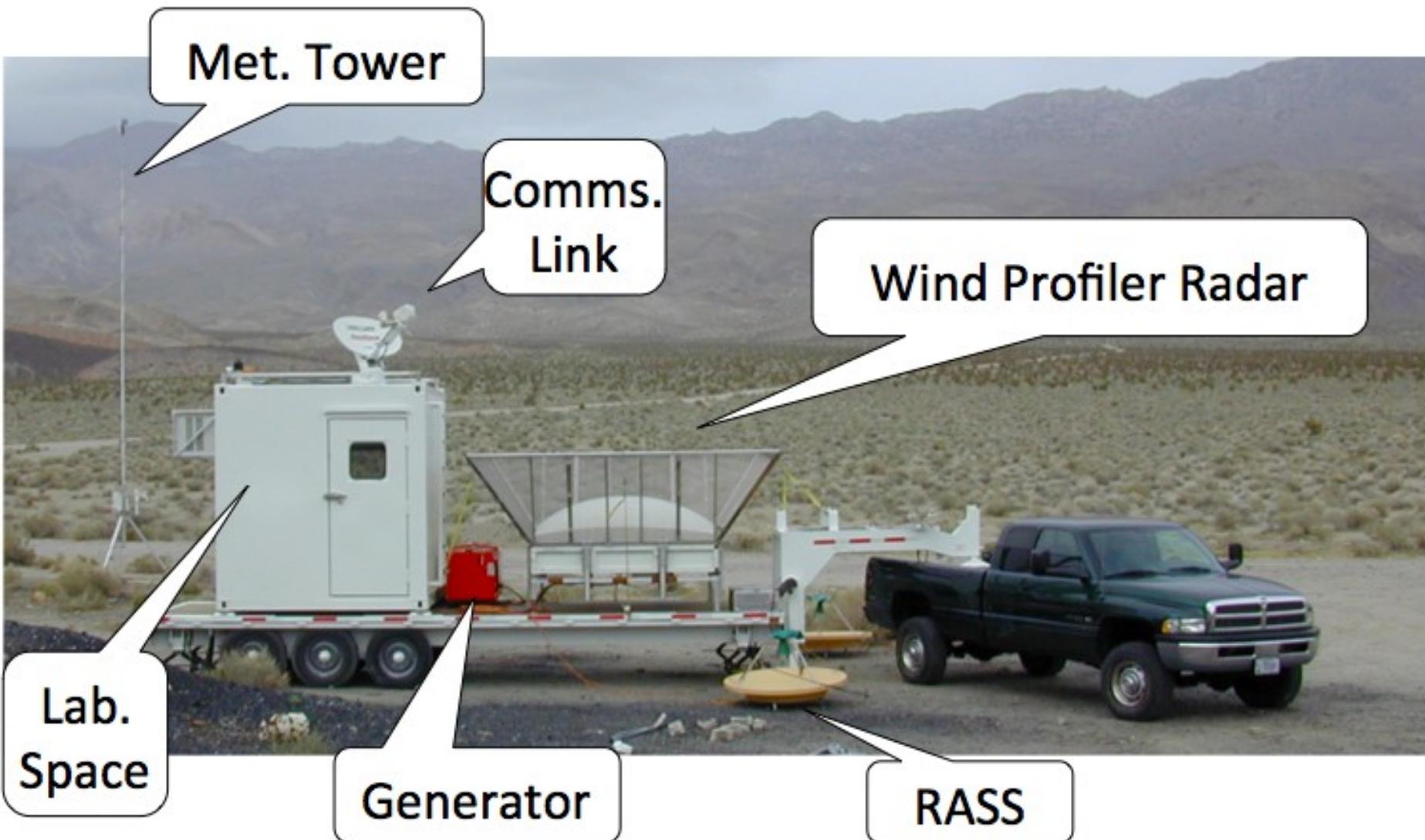
Integrated Sounding System

The ISS is a **platform** for hosting a collection of boundary-layer **instruments** which can be **configured** for each field project site while sharing a common deployment and **data** management infrastructure, comprised of **networking**, computer **systems**, and **software**.

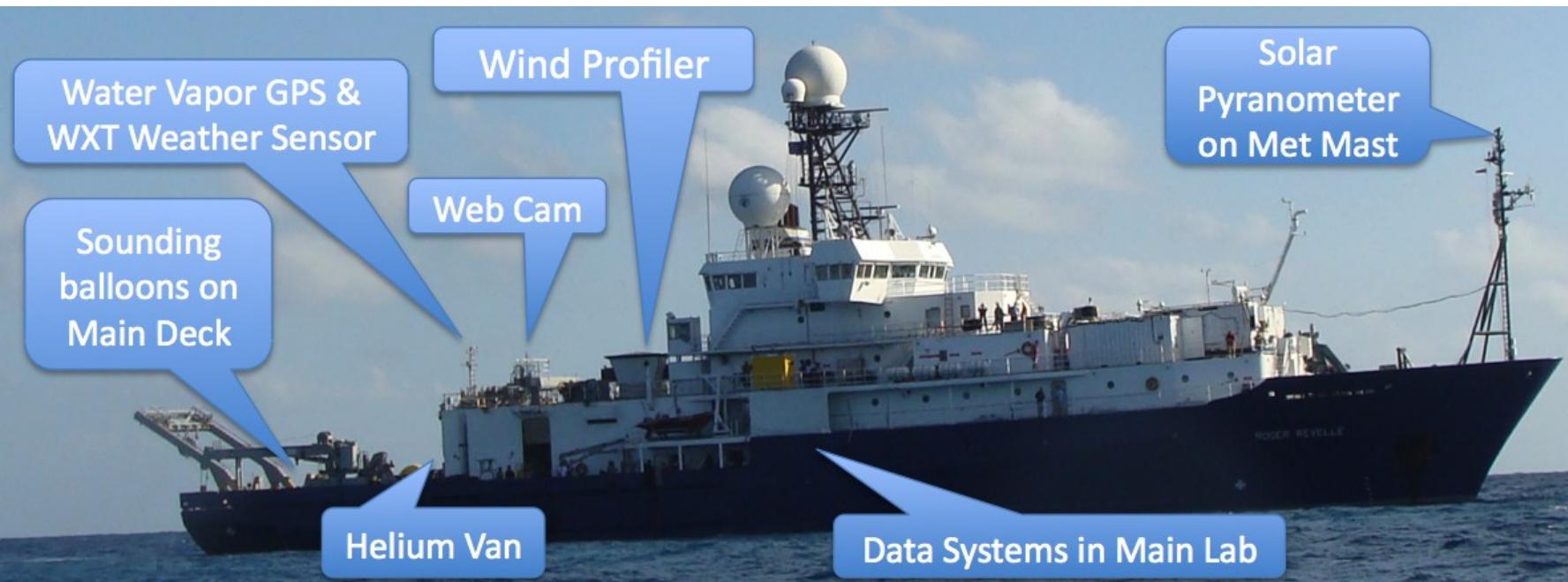
ISS at DYNAMO, Diego Garcia



Mobile ISS at T-REX



ISS at DYNAMO, R/V Revelle



Technical Debt

Shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite. ... The danger occurs when the debt is not repaid.

- Ward Cunningham

Technical Debt

Shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite. ... The danger occurs when the debt is not repaid.

- Ward Cunningham

All field project deployments are like shipping first time code.

- Me

Technical Debt

Every minute spent on not-quite-right code counts as interest on that debt. Entire engineering organizations can be brought to a stand-still under the debt load of an unconsolidated implementation, object- oriented or otherwise.

- Ward Cunningham

Technical Debt Pressures in the Field

- Mission Creep
 - Since we're so close already...
- Live Testing
 - Now that we've got everything all together
- User Interaction
 - By the way, now I understand what I really need...
- Thrill and Excitement
 - What's more fun than real-time coding under pressure?

SABL LIDAR on the C-130



Other Technical Debt in the Field

- Instrument Data
 - If you record it, you own it
- System Configuration
 - Once you tweak it, you have to keep tweaking it
- Issue Documentation
 - “Those who forget the past are doomed to repeat it.”
- Users never forget a feature
 - Adding a feature in the field is never a one-time thing
- Outside instruments and software
 - Paying interest on debt someone else accumulated

ISS as a Test Case

(Putting Debt Reduction into Practice)

Reducing Technical Debt in the Field (1)

- Design to Deploy
 - Software is not working until it is deployable
 - Python classes wrap external, third-party programs
- Test Deployments
 - Deploy tests and run them on targets
 - Script installation and confirmation of dependencies

Reducing Technical Debt in the Field (2)

- Consolidate and deploy the configuration
 - Single access point
 - No redundant configuration
 - Keep it in revision control
 - Beware overlooked configuration sources like crontabs, BIOS settings, routers, web devices
- Use the configuration for devops also

System Configuration

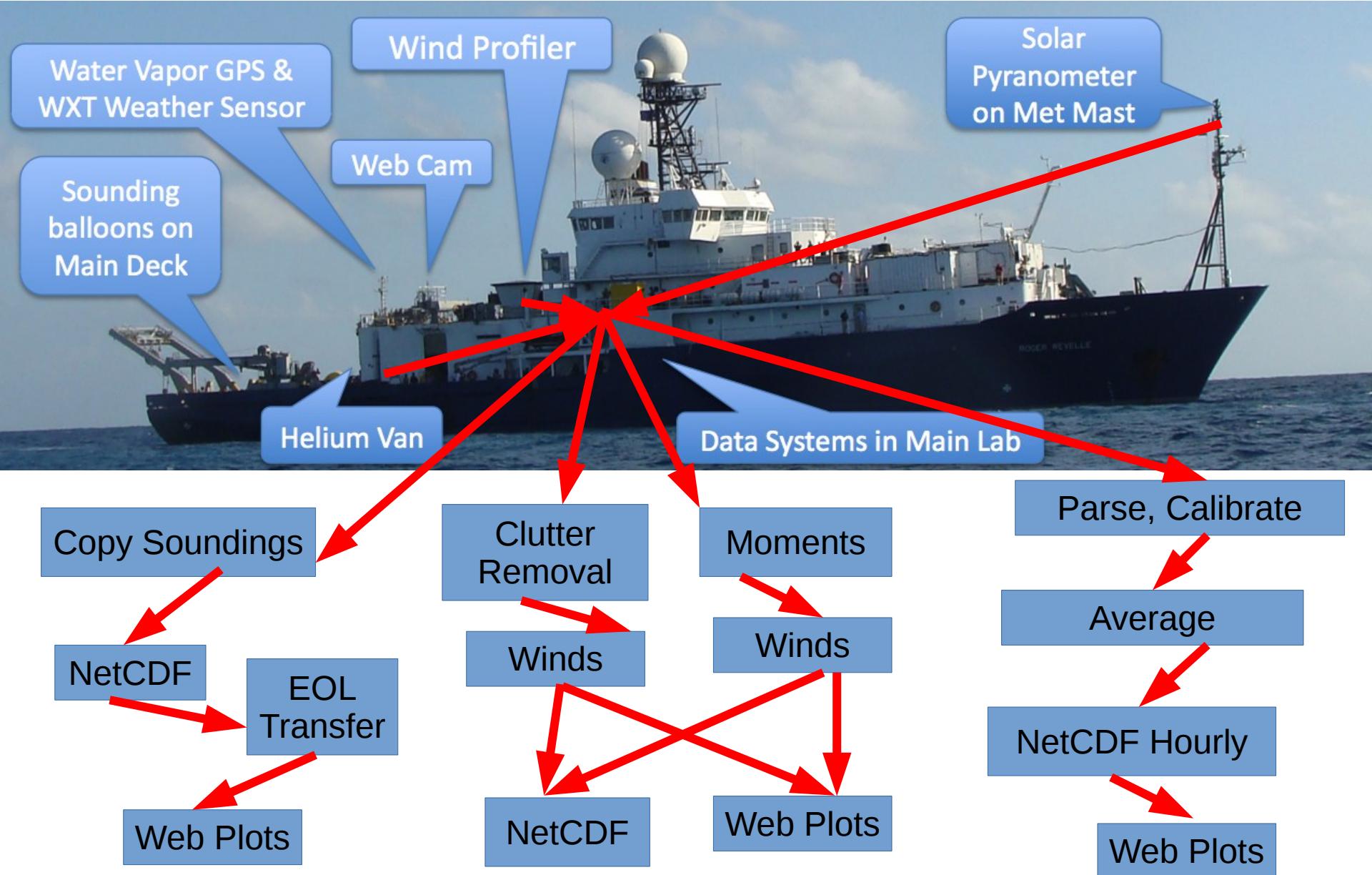
- Bare metal installs and restores with kickstart files and bash scripts
- Modular bash scripts for configuring every instrument data flow (slowly migrating to python)
- JSON configuration file for each site kept in revision control
- Scripts can be re-run at any time to update the system as the configuration changes

Reducing Technical Debt in the Field (3)

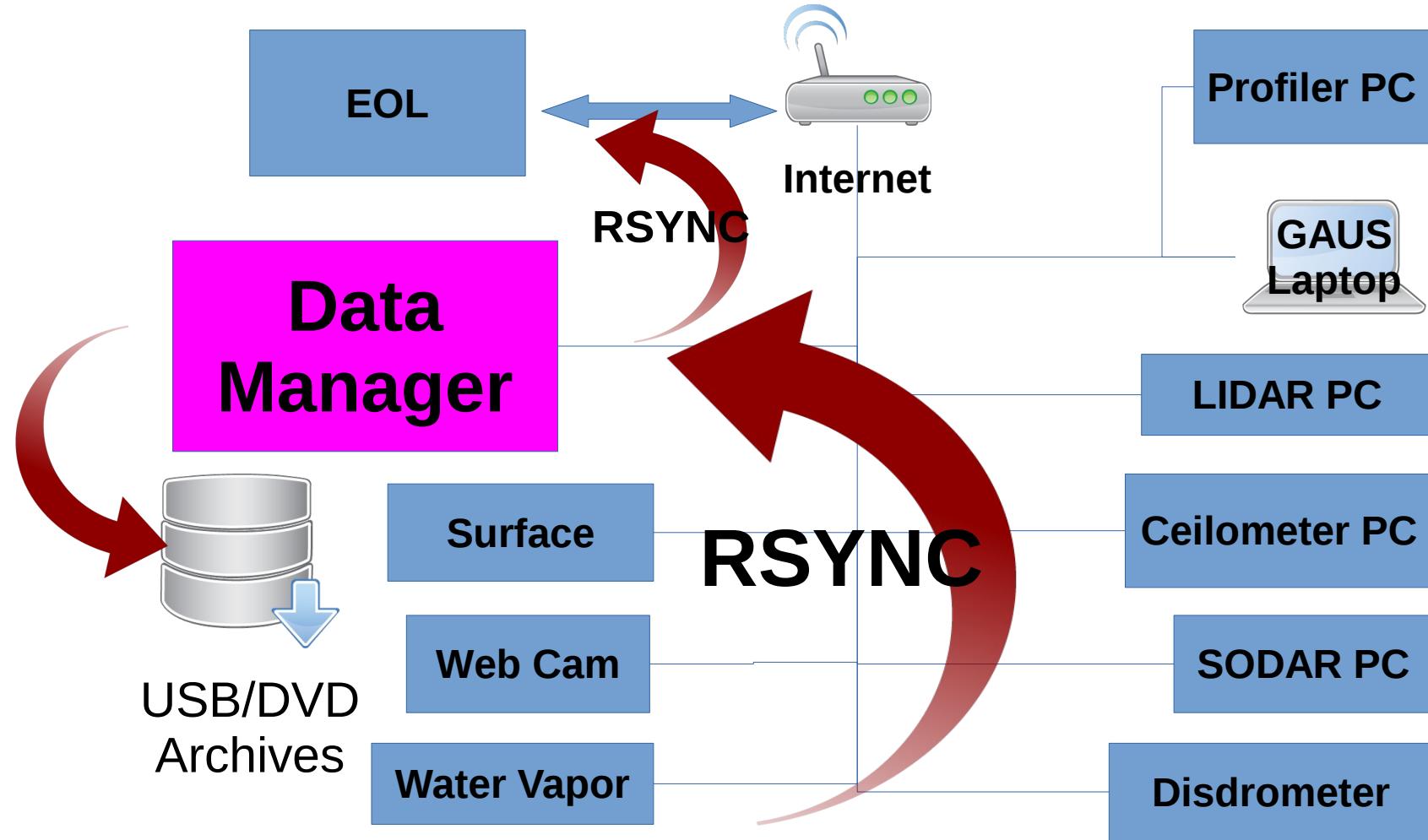
- Automate failure detection
 - Whatever checks you do manually, script the ones that can be scripted for monitoring tools like Nagios
 - For ISS, custom python scripts check USB backups, data broadcasts, missing data

Applying Devops to Data: Data-Ops

- Treat data streams like devops servers
 - Easy to deploy from one field project to the next
 - Easy to add and manage new data streams
- Automate deployment of the tools needed to process each data stream
- Continuously generate deliverable data products from the raw data streams and a single project configuration
 - Like continuous integration for software, but for data



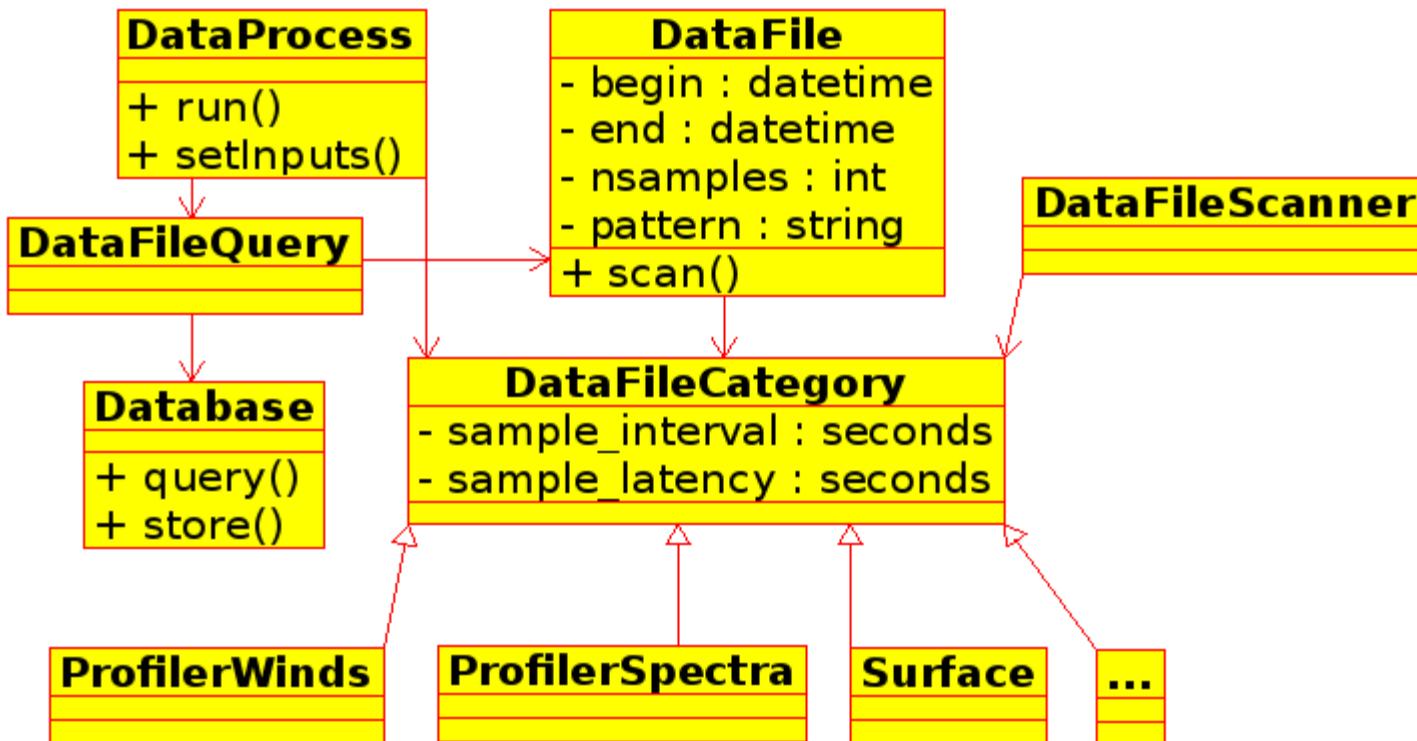
ISS Field Network



ISS Python Catalog

- Scan: filename patterns, simple format parsing including netcdf and SPC, read data samples
- Catalog: query by name, time, category, size, checksums, and more complicated
- Check: gaps, latency, sample density, custom, nagios reports, email alerts
- Archive: ISO image with metadata from catalog query, integrity checks, transfers to EOL
- Derive: Run processing when inputs have changed

ISS_Catalog Class Diagram



Python Helps

- Parsing of custom data files, text and binary
- Scipy.io.netcdf and numpy
- Built-in sqlite
- Built-in JSON
- py.test and pylint
- Potential evolution into a web app

Dev and Data Ops

Goals for ISS:

- Configure an entire project deployment, all of its sites, instruments, data streams, derivations, and monitoring, with just the JSON parameter database and a python customization script.
- Implement the real-time and post-project data processing like a continuously running build process, where all derived data products are regenerated and updated as the raw data files are filled in or as project parameters change.

Conclusion

Technical debt is a useful metaphor for software engineers and maybe even managers.

Consolidate and deploy the project configuration.

Automate field deployments using devops and based on the project configuration.

Automate data processing with parameters from the project configuration.

Integrate and automate with python. (rsync is awesome too.)