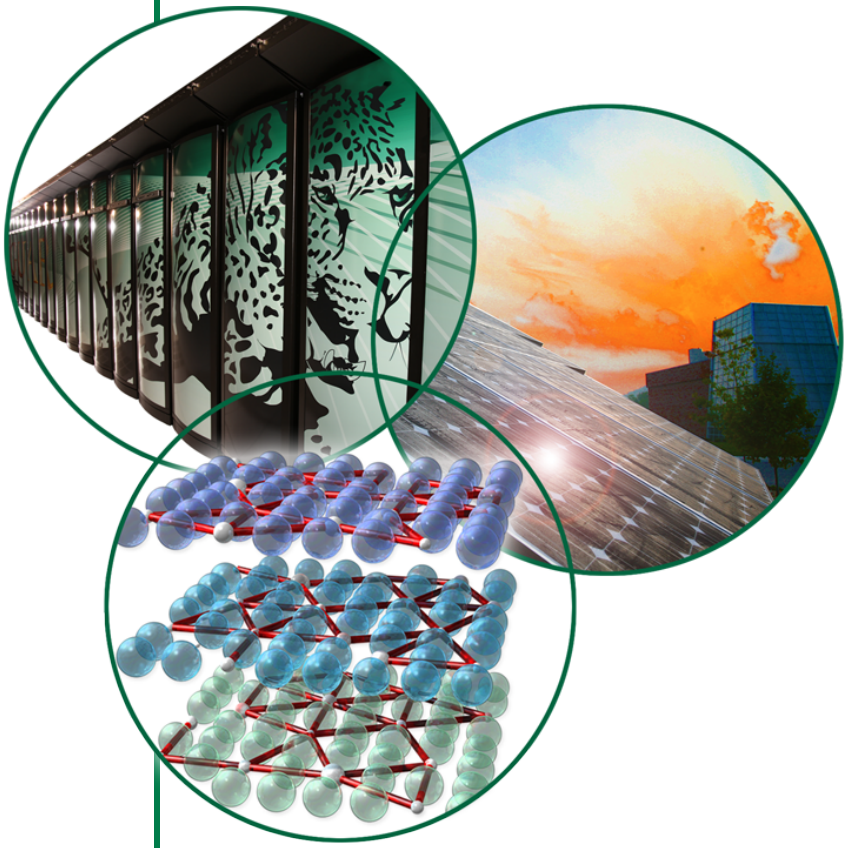
# How to Automate Neutron Scattering Experiment with Python?

**Piotr Żołnierczuk, Richard Riedel**

**Neutron Scattering Science Division**
**Oak Ridge National Laboratory**



- **Introduction & Motivation**
- **PyDas Overview**
- **Summary**

U.S. DEPARTMENT OF ENERGY

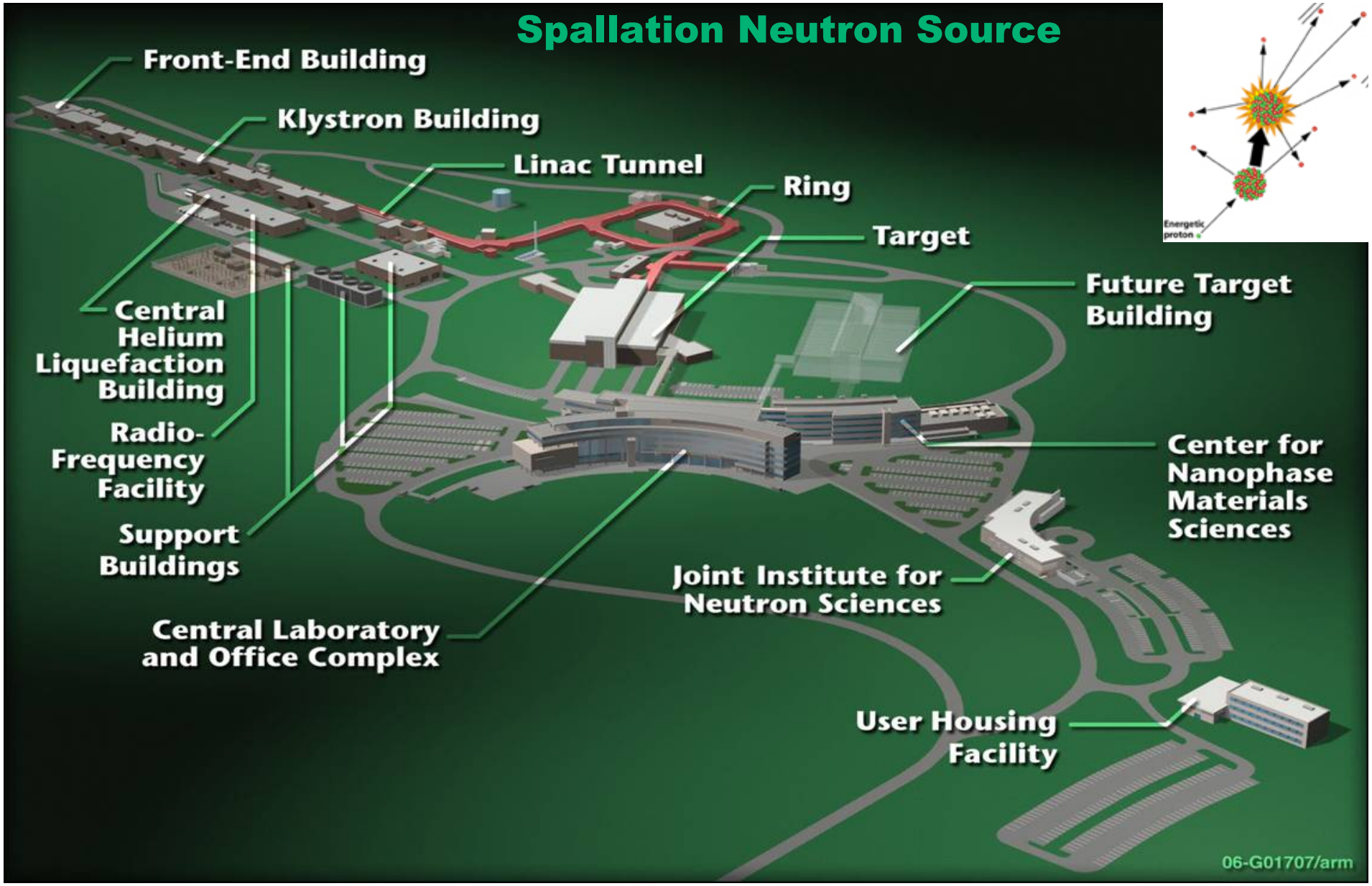
***NOBUGS 2010, Gatlinburg, TN***

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

# Update: From NOBUGS 2008

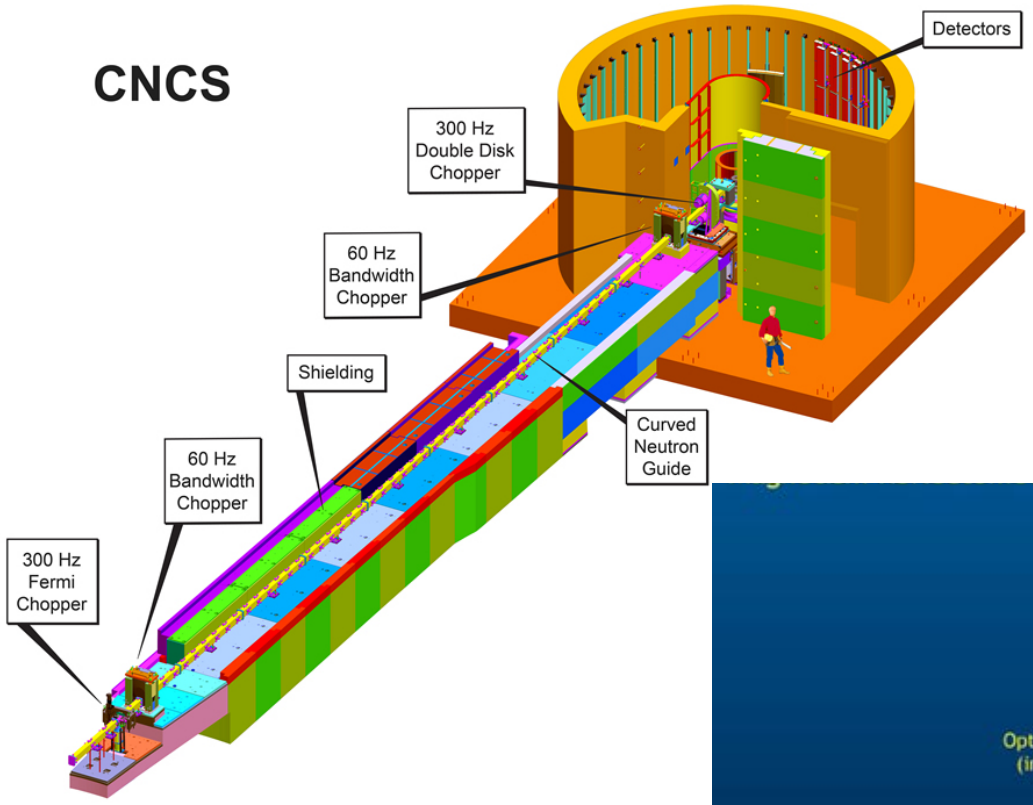
# Update: To NOBUGS 2010

# Spallation Neutron Source

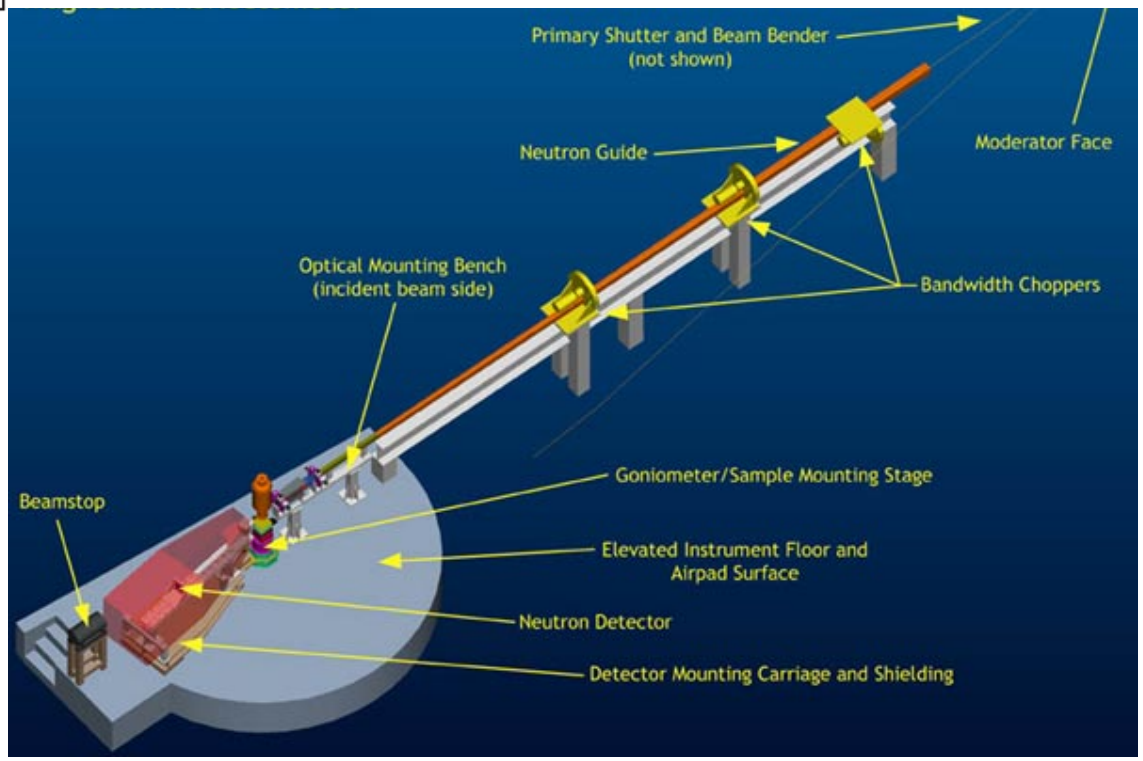


# SNS Instruments

## CNCS

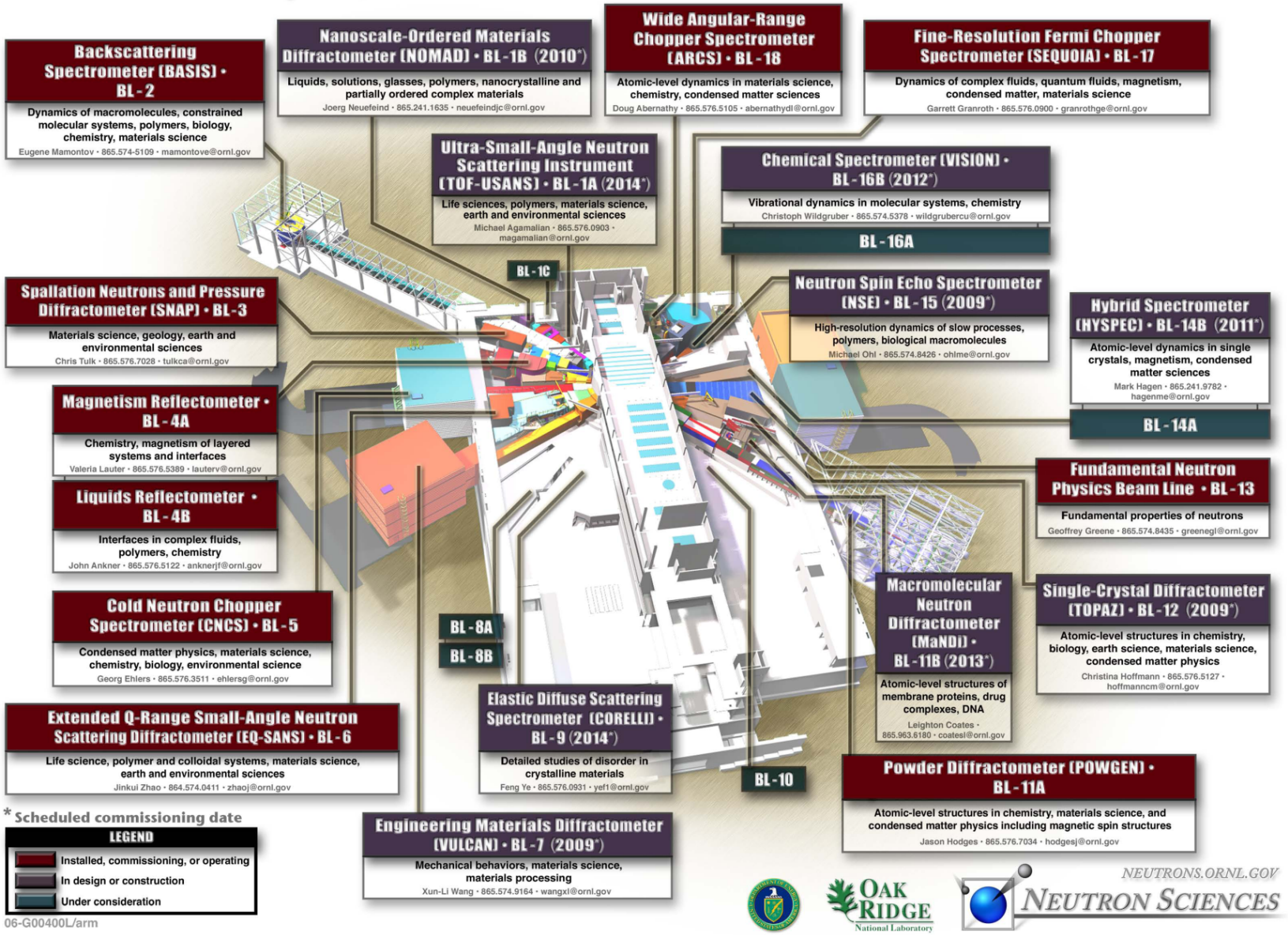



## REF-M

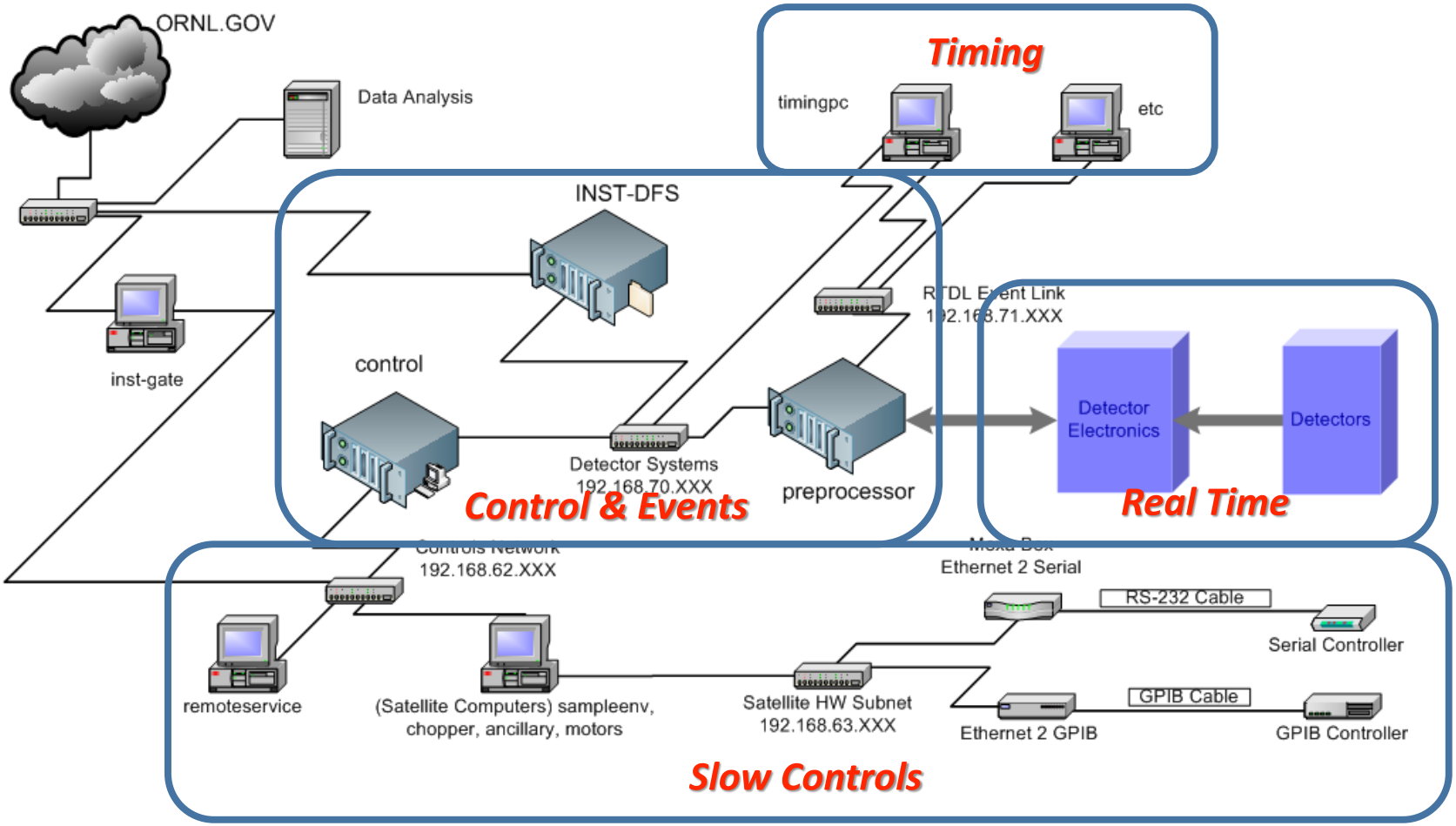


# Spallation Neutron Source at Oak Ridge National Laboratory

**The world's most intense pulsed, accelerator-based neutron source**

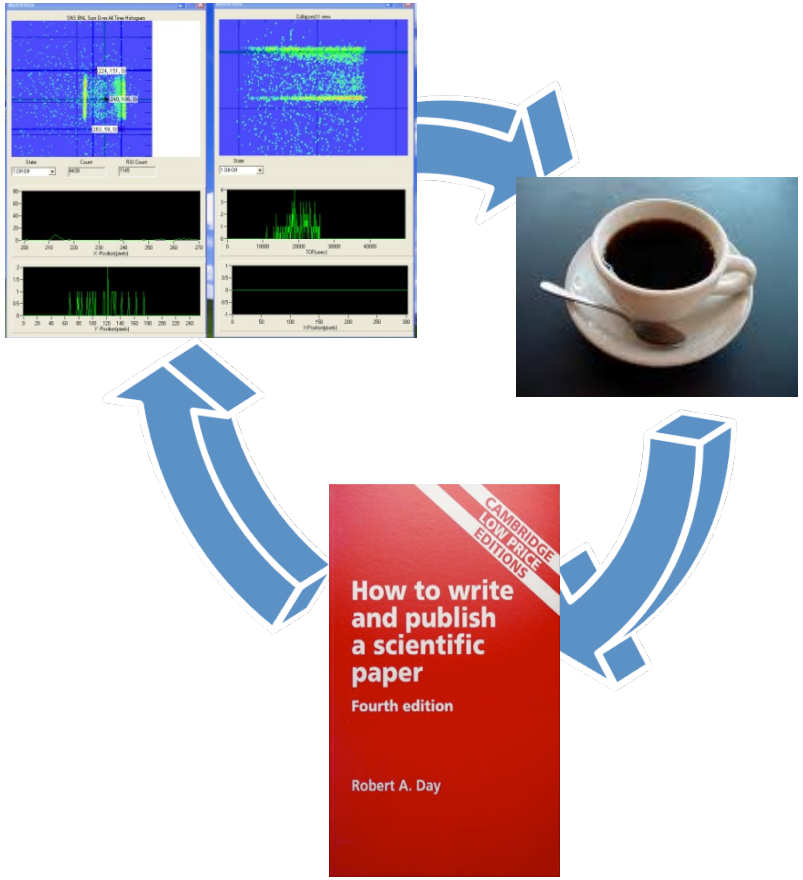


# SNS DAS Architecture




See also **Rick Riedel** *UDP, TCP, Circular Buffers, Multi-Threaded Programming and the Transmission of Event Data* (Tuesday at Best Practices)

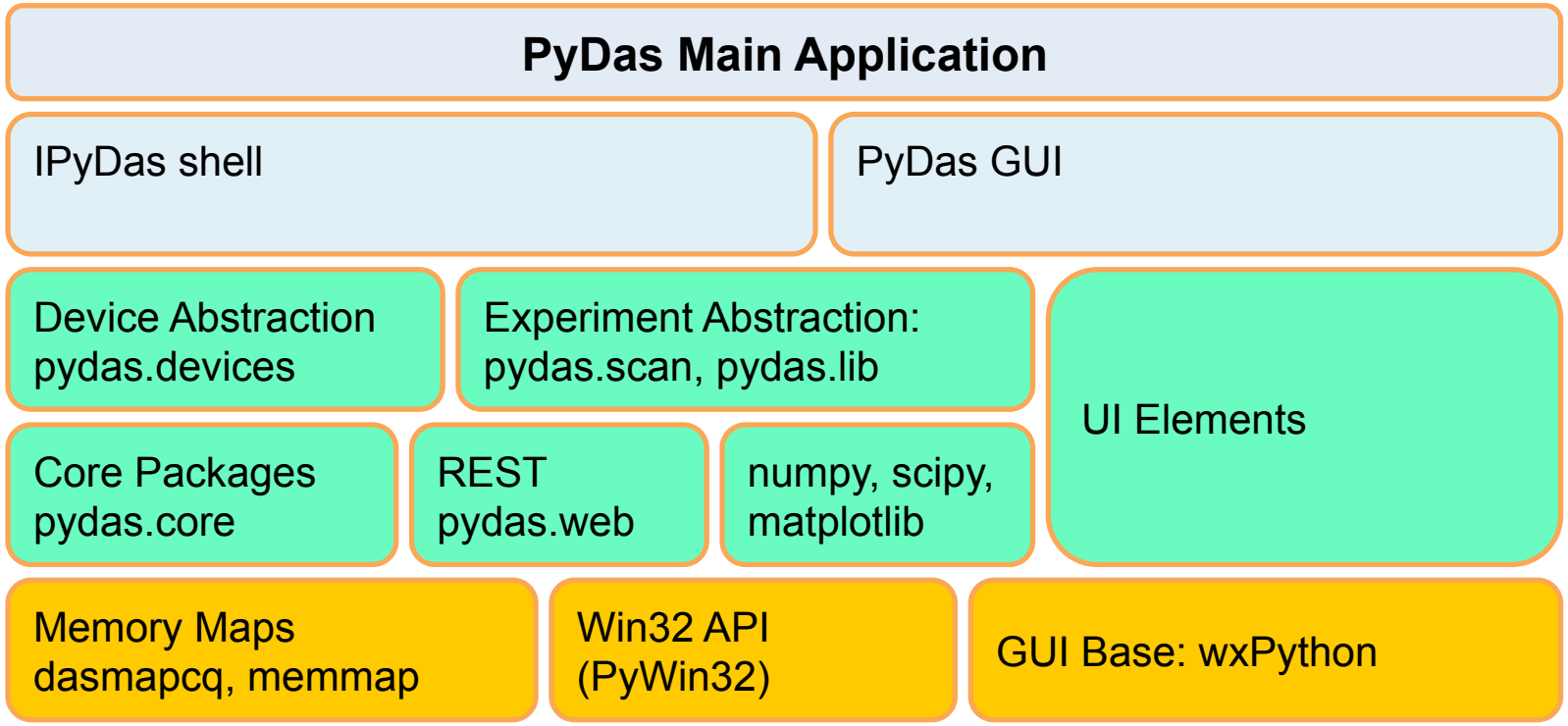
# Why have we done this?



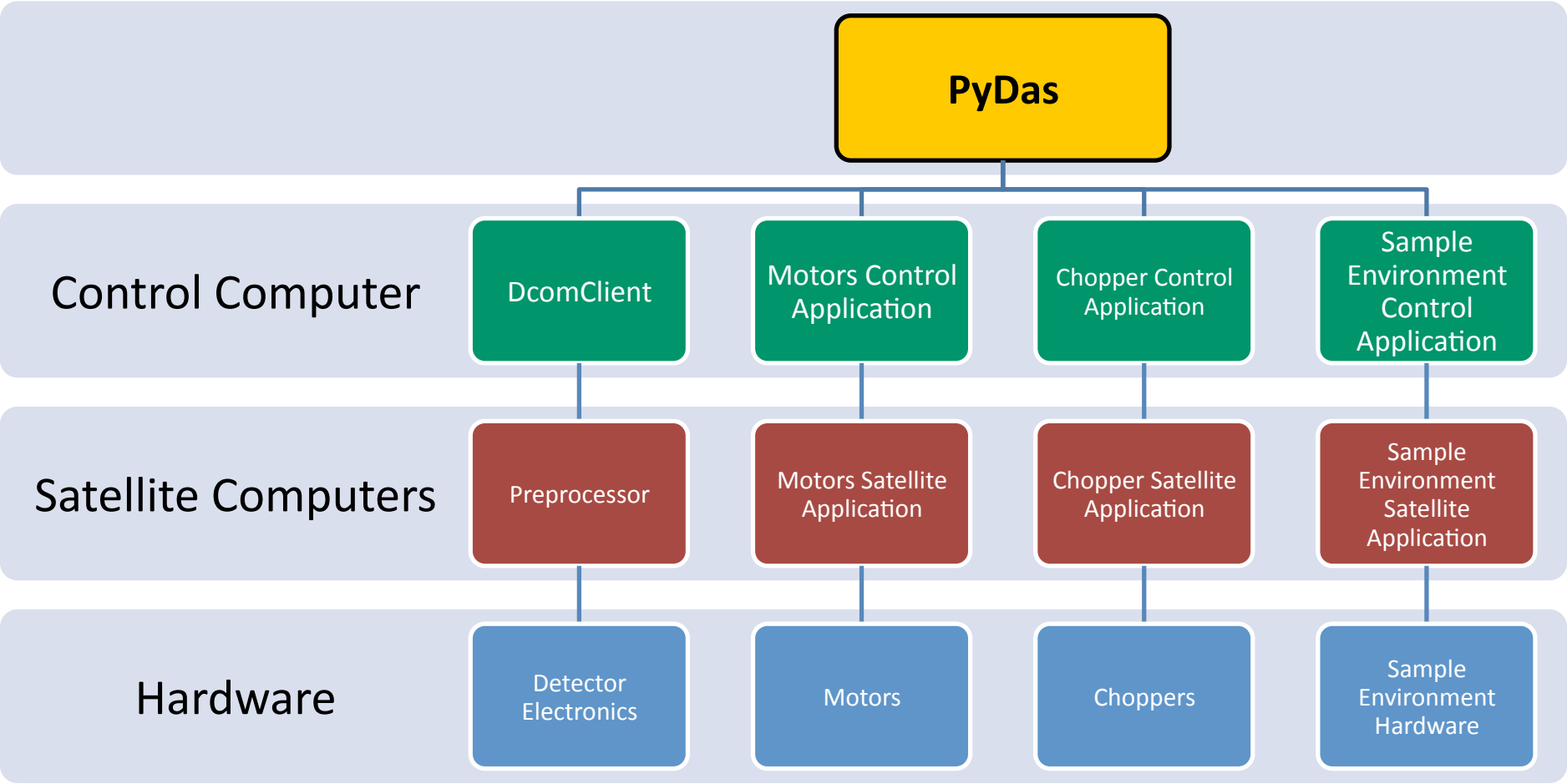
# PyDas History

- **Project started as a scripting "hook" to the SNS DAS**
  - **soon Instrument Scientists wanted more**
- **Initially each instruments was a separate project**
  - **Maintenance & support were major issues**
  - **We still use the "old " at three beam lines ☹**
- **For new beam lines we consolidated the project new PyDas ☺**
  - **Customization via configuration files and beam line specific modules**

# PyDas Architecture



# PyDas Communication Overview



# Examples: Scanning in IPyDas

```
scan('motor1',numpy.arange(10.0,90.0,0.5),
            RUNTIME, 30.0, go_back=True, plot='counts')
```

```
for SampleTemp.value in [273.0,293.0,313.0,333.0]:
     scan('Energy', [25.,35.,50], PCHARGE, 1e12,
            title='Nobel Prize Data at T=%s K' % temp)
```

```
tscan(‘myscan.csv‘)
```

# Examples: Table scan

- ***Spreadsheet file example***

```
#Title:    , "My Scan"
#Comment: , "comments are good"
X, Y, Z, Phase1, Slit2, CountFor, CountValue,  Comment
1, 2, 4, 30,          3, pcharge , 1e12       , "set point 1"
 , 3,  , 30,           , runtime , 3600       , "set point 2"
```

- ***File format***
  - ***CSV (via Python module csv)***
  - ***Or any other format that Gnumeric [1] can convert to CSV (e.g. Excel, OpenOffice)***

1) http://projects.gnome.org/gnumeric/

# Plotting in PyDas

- **Use matplotlib**
  - http://matplotlib.sourceforge.net
- **Take (almost) everything literally from the pyplot tutorial**
  - **Do not import `pyplot`**
  - **Replace `plt` with `fig`**
  - **Only single figure at the moment**

# Some other features

- **Simple fitting**

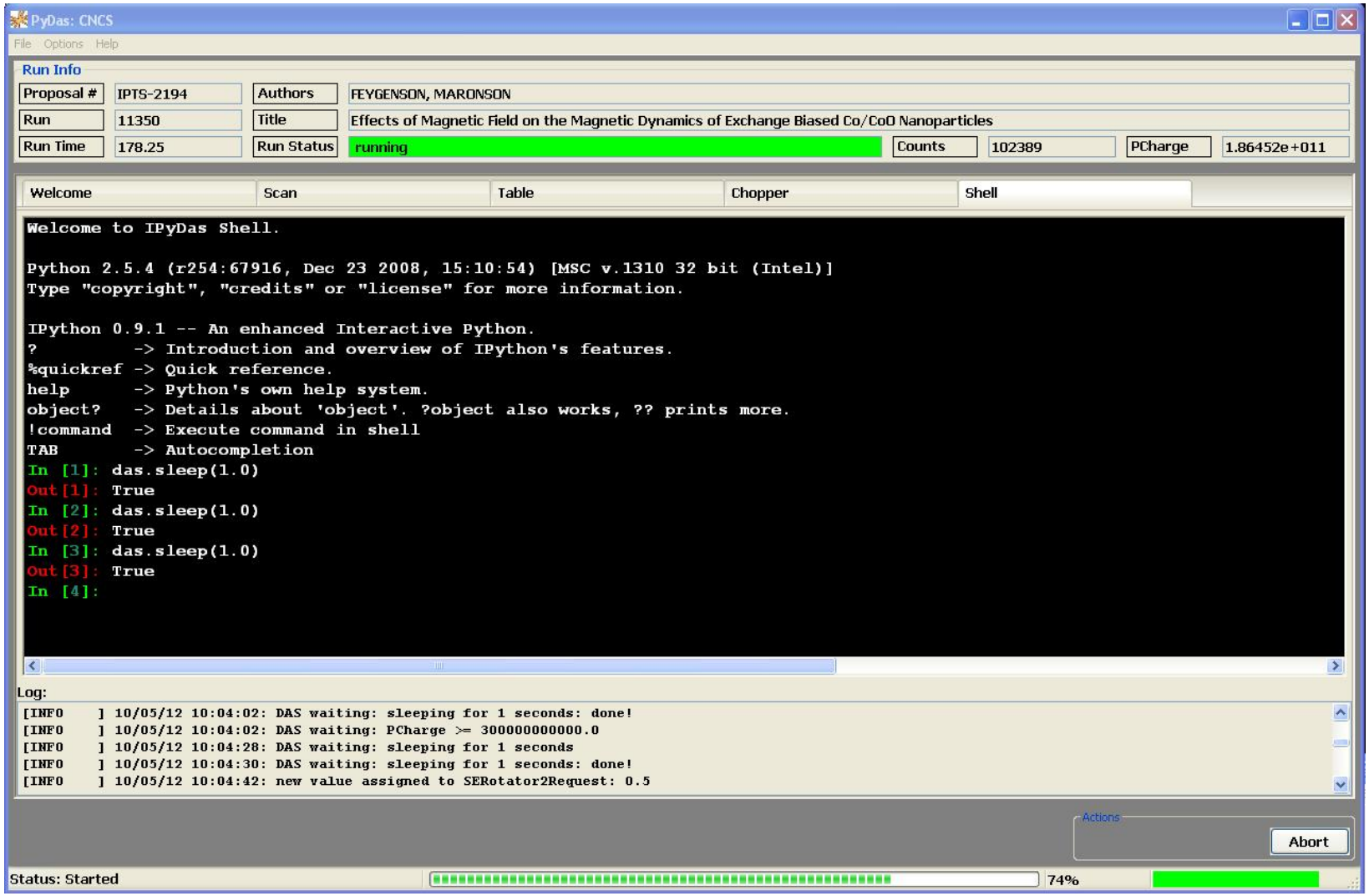
```
s = scan('m1',range(0,10,1), PCHARGE, 1e10, plot='counts')
s.fit('gauss+poly2')
```

- **RESTful Web Interface**

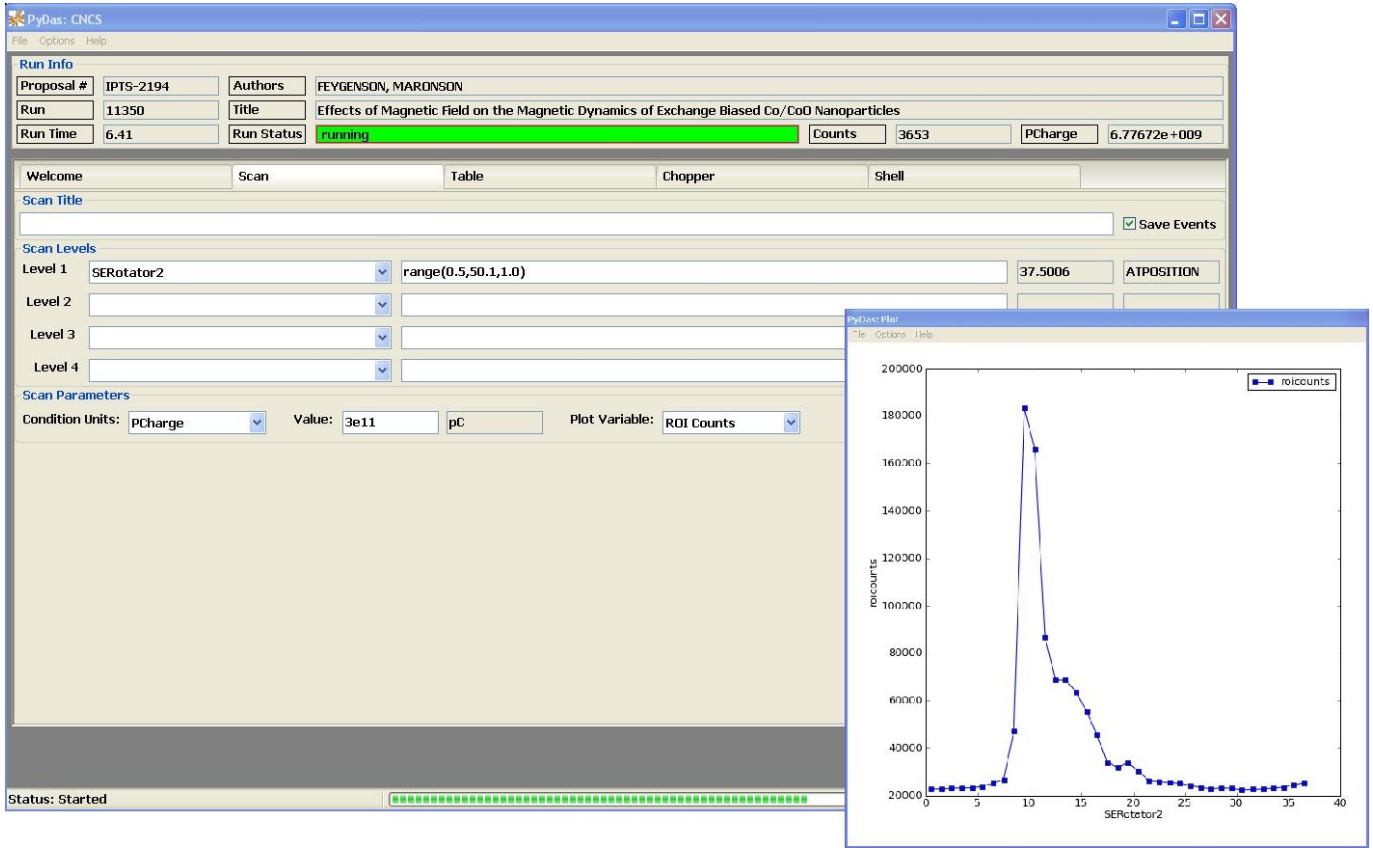
```
cli = DasWebClient('http://remotehost:8080/QOQ')
cli.wait()
newpoints = cli.get('/PTS')
```

More: **Cathy Jiao**, RESTful Web Services for Real-Time Neutron Scattering (Poster Session)

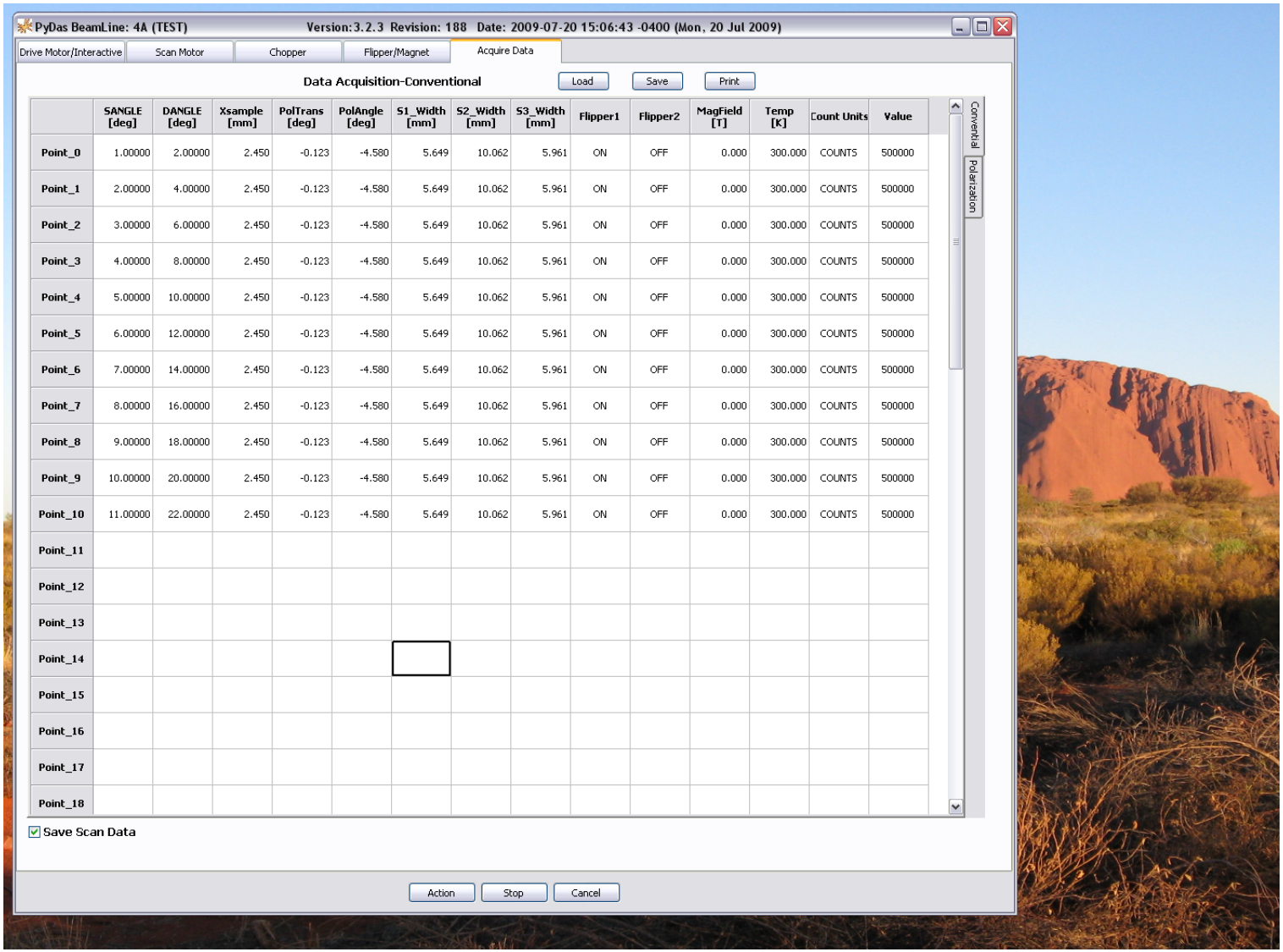
# PyDas Shell (IPyDas)



# PyDas GUI Examples



# Legacy PyDas in Action




| | SANGLE [deg] | DANGLE [deg] | Xsample [mm] | PolTrans [deg] | PolAngle [deg] | S1_Width [mm] | S2_Width [mm] | S3_Width [mm] | Flipper1 | Flipper2 | MagField [T] | Temp [K] | Count Units | Value |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Point_0 | 1.00000 | 2.00000 | 2.450 | -0.123 | -4.580 | 5.649 | 10.062 | 5.961 | ON | OFF | 0.000 | 300.000 | COUNTS | 500000 |
| Point_1 | 2.00000 | 4.00000 | 2.450 | -0.123 | -4.580 | 5.649 | 10.062 | 5.961 | ON | OFF | 0.000 | 300.000 | COUNTS | 500000 |
| Point_2 | 3.00000 | 6.00000 | 2.450 | -0.123 | -4.580 | 5.649 | 10.062 | 5.961 | ON | OFF | 0.000 | 300.000 | COUNTS | 500000 |
| Point_3 | 4.00000 | 8.00000 | 2.450 | -0.123 | -4.580 | 5.649 | 10.062 | 5.961 | ON | OFF | 0.000 | 300.000 | COUNTS | 500000 |
| Point_4 | 5.00000 | 10.00000 | 2.450 | -0.123 | -4.580 | 5.649 | 10.062 | 5.961 | ON | OFF | 0.000 | 300.000 | COUNTS | 500000 |
| Point_5 | 6.00000 | 12.00000 | 2.450 | -0.123 | -4.580 | 5.649 | 10.062 | 5.961 | ON | OFF | 0.000 | 300.000 | COUNTS | 500000 |
| Point_6 | 7.00000 | 14.00000 | 2.450 | -0.123 | -4.580 | 5.649 | 10.062 | 5.961 | ON | OFF | 0.000 | 300.000 | COUNTS | 500000 |
| Point_7 | 8.00000 | 16.00000 | 2.450 | -0.123 | -4.580 | 5.649 | 10.062 | 5.961 | ON | OFF | 0.000 | 300.000 | COUNTS | 500000 |
| Point_8 | 9.00000 | 18.00000 | 2.450 | -0.123 | -4.580 | 5.649 | 10.062 | 5.961 | ON | OFF | 0.000 | 300.000 | COUNTS | 500000 |
| Point_9 | 10.00000 | 20.00000 | 2.450 | -0.123 | -4.580 | 5.649 | 10.062 | 5.961 | ON | OFF | 0.000 | 300.000 | COUNTS | 500000 |
| Point_10 | 11.00000 | 22.00000 | 2.450 | -0.123 | -4.580 | 5.649 | 10.062 | 5.961 | ON | OFF | 0.000 | 300.000 | COUNTS | 500000 |
| Point_11 | | | | | | | | | | | | | | |
| Point_12 | | | | | | | | | | | | | | |
| Point_13 | | | | | | | | | | | | | | |
| Point_14 | | | | | | | | | | | | | | |
| Point_15 | | | | | | | | | | | | | | |
| Point_16 | | | | | | | | | | | | | | |
| Point_17 | | | | | | | | | | | | | | |
| Point_18 | | | | | | | | | | | | | | |

# Best Practices

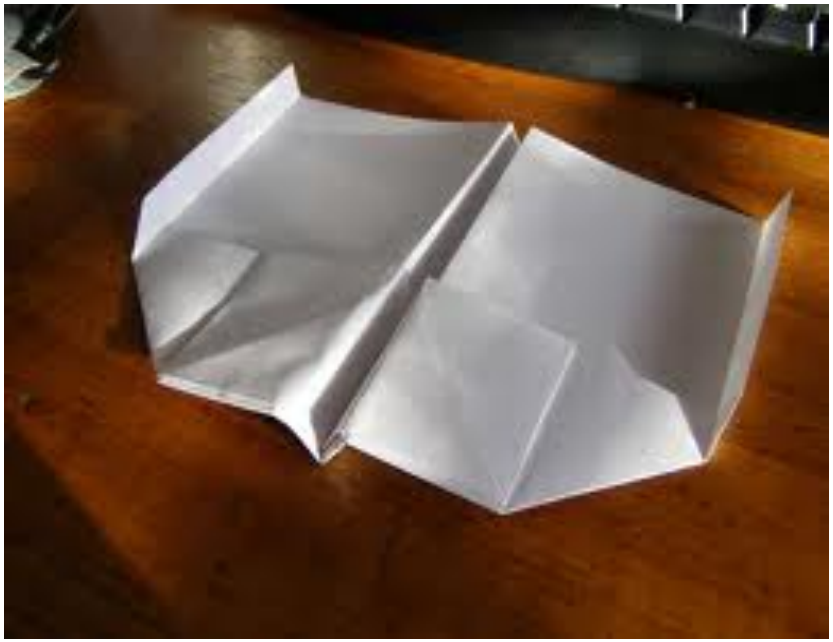
***NOBUGS 2010 Web Site:***

*… this time we try to do something different. We wish to put NOBUGS 2010 under the general theme "Best Practices". We want you to share your experiences. Thus we encourage you to enter presentations which follow the theme: this is what we tried, this is what we learned, this is what worked for us, this failed! A good example of what we are after is the talk given by Paul Lewis at NOBUGS 2008 in Sydney. We envisage that we arrange contributions of this style into appropriate sessions followed by a discussion in which we work out the best practices as a guide for newcomers and managers. The program committee will prefer presentations following this pattern.*

# Test Environment – Mock Objects

**Real Object**

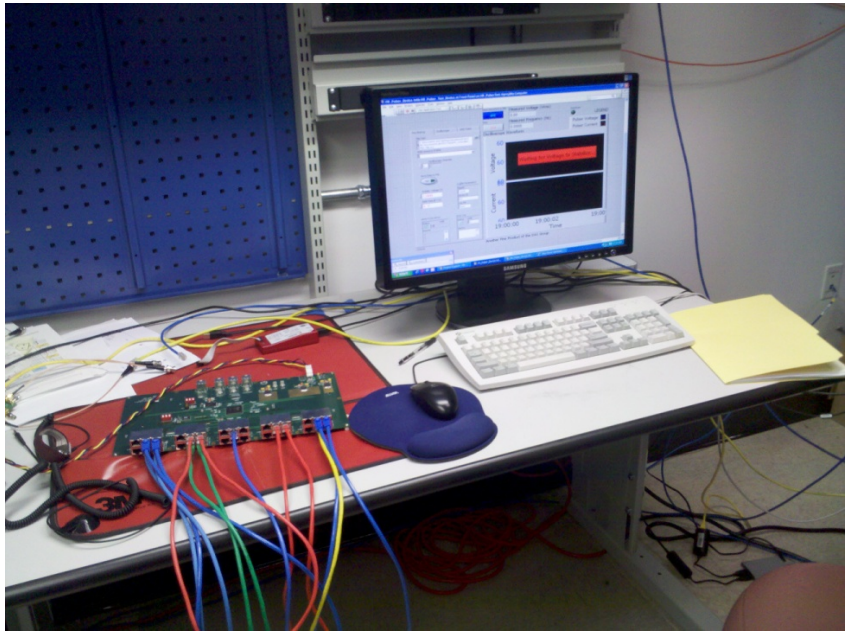
**Mock Object**

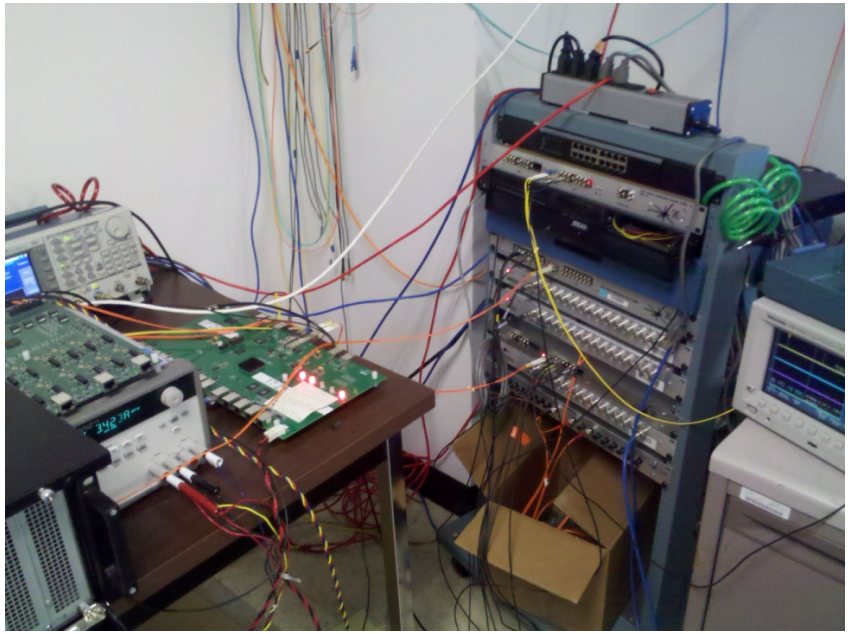


# Test Environment – Unit Testing with Mock Objects

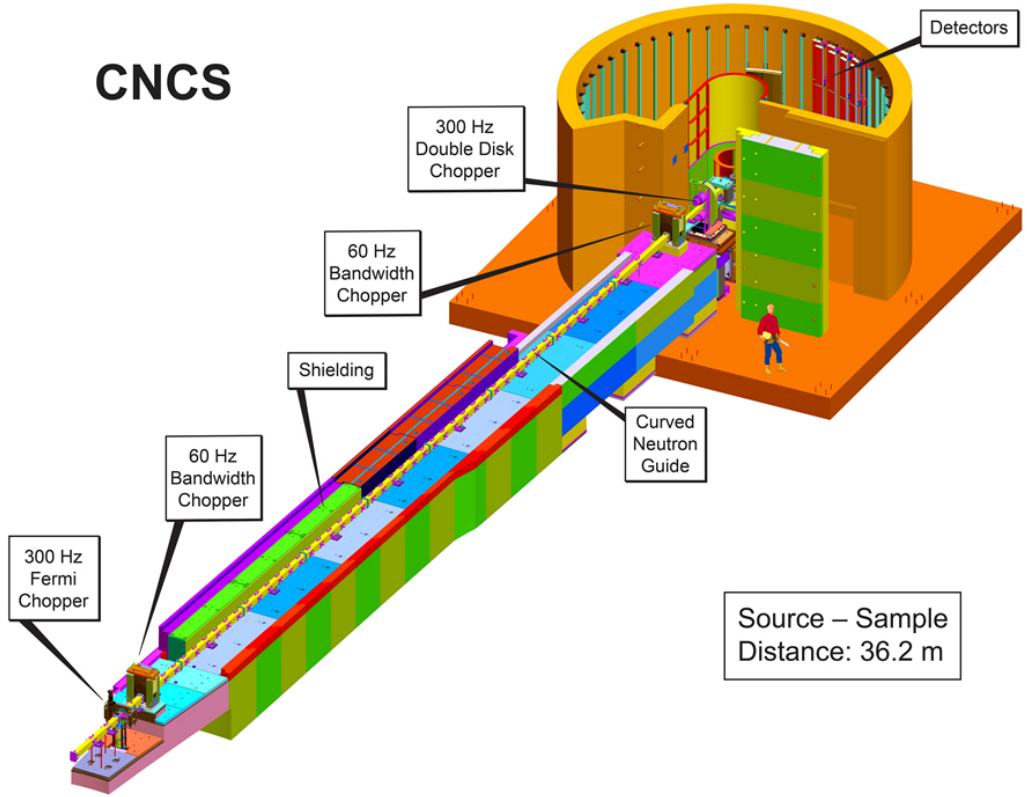
```
import unittest
import pydas.devices          as devices
import pydas.core.snsnake     as snsnake
import pydas.core.mocksnake   as mocksnake

class MotorTestCase(unittest.TestCase):
    "Test cases for Motor class"

    def setUp(self):
        "Unit test set-up method"
        mocksnake.create_vars(devices.Motor , 'motor1')


    def testValueProperty(self):
        "Motor value property test case"
        mot1 = devices.Motor('motor1', units='deg')
        mot1.value = 2.0

        self.assertEquals(mot1.value  , 2.0)
        self.assertEquals(mot1.request, 2.0)
        self.assertEquals(mot1.status , 'ATPOSITION')
        self.assertEquals(mot1.units  , 'deg')
```

# Test Environment – Test Beam Line



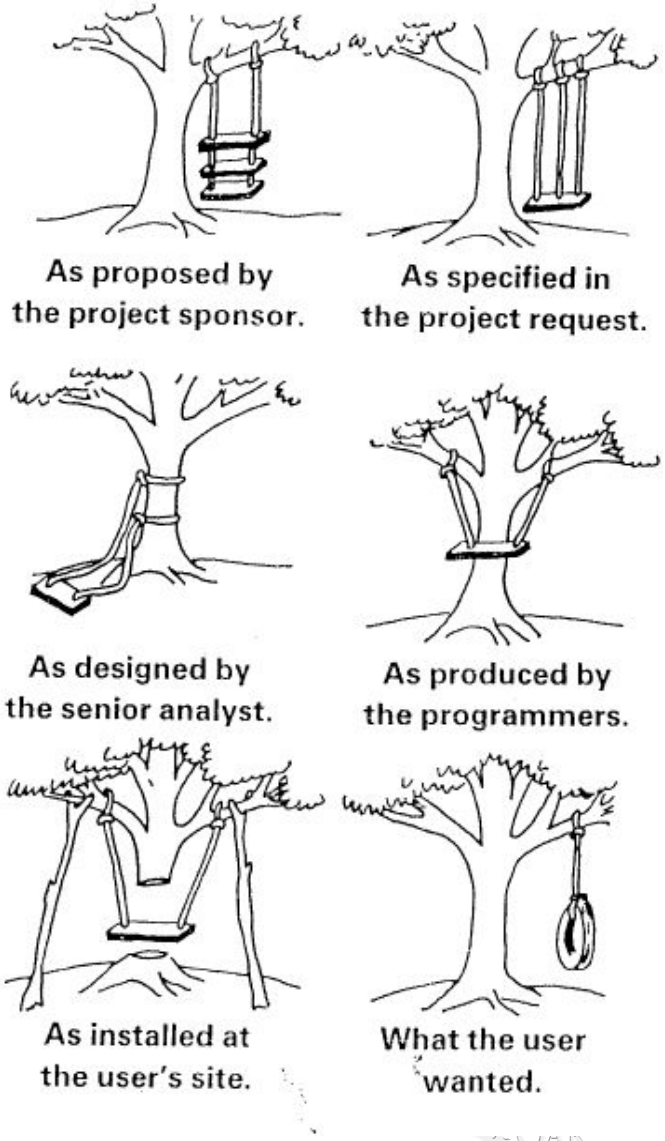


# Test Environment – Real Instrument



# Summary & Outlook

- **PyDas "glues" various DAS components together**
- **need to support 24 different instruments**
- **testing, testing, and more testing**
- **exploring IronPython/.NET: see Madhan Sundaram poster** *Das4Net: Next Generation Das Suite of Software*




Thanks to all engineering staff from NSS DAS Group
Lloyd Clonts, Gayle Greene, Steve Hicks, Andre Parizzi, Richard Riedel, Mariano Ruiz-Rodriguez, Madhan Sundaram and Wenduo Zhou