Data Management at Alba

...or "how we acquire and write data with Sardana"

Carlos Pascual-Izarra
(on behalf of Alba BL controls group)

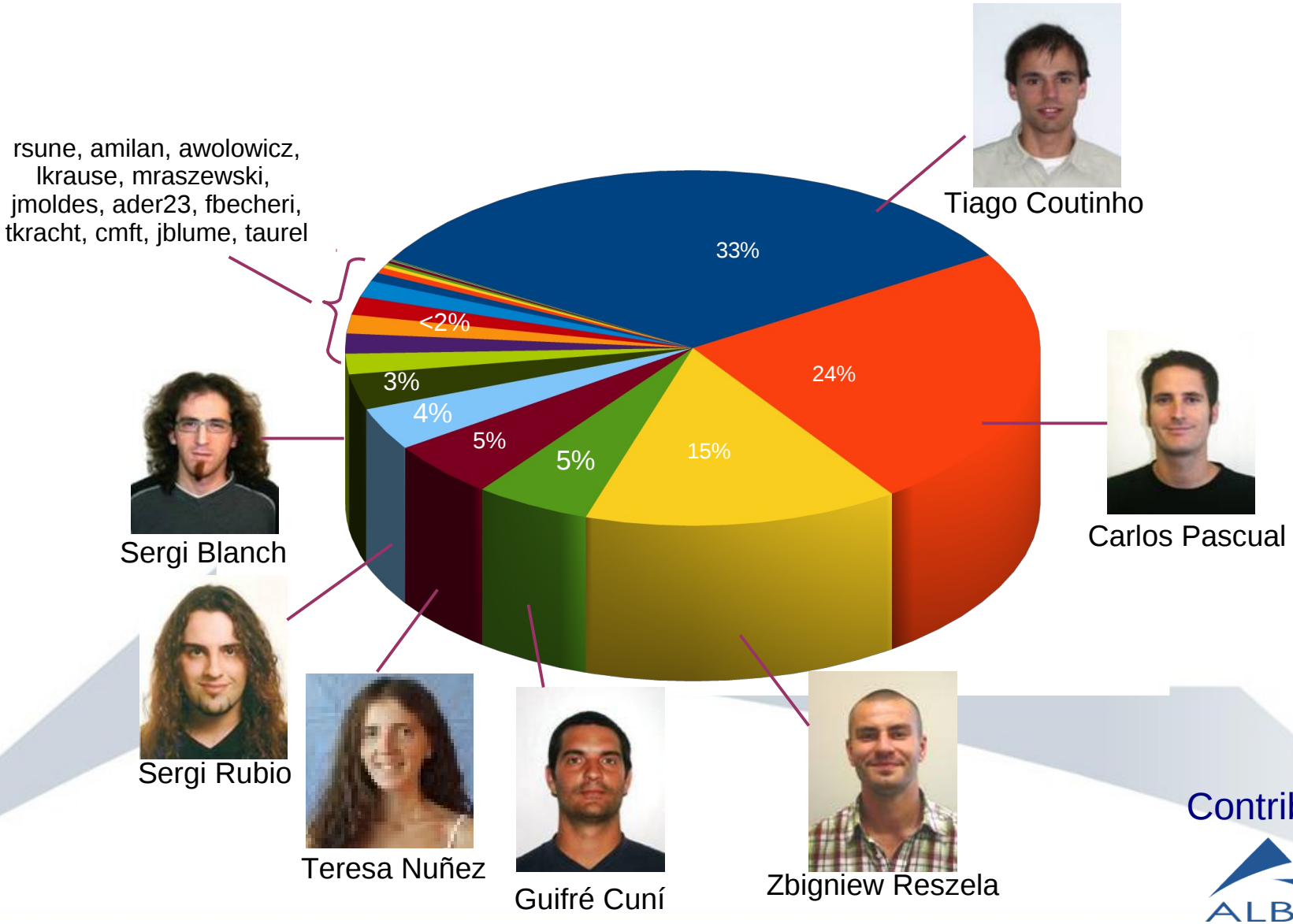
NOBUGS -2012
Rutherford Appleton Laboratory, UK
24-26 of September 2012

Overview

- Sardana as seen by the users:
  - Command line interface (Spock)
  - Graphical user interface (Taurus)
- Data handling (Pool and Macroserver):
  - DataHandler & data recorders
  - NXscan file recorder
  - NXxas file recorder
- Pending issues

Contributors to Sardana

svn commits by author to sardana & taurus



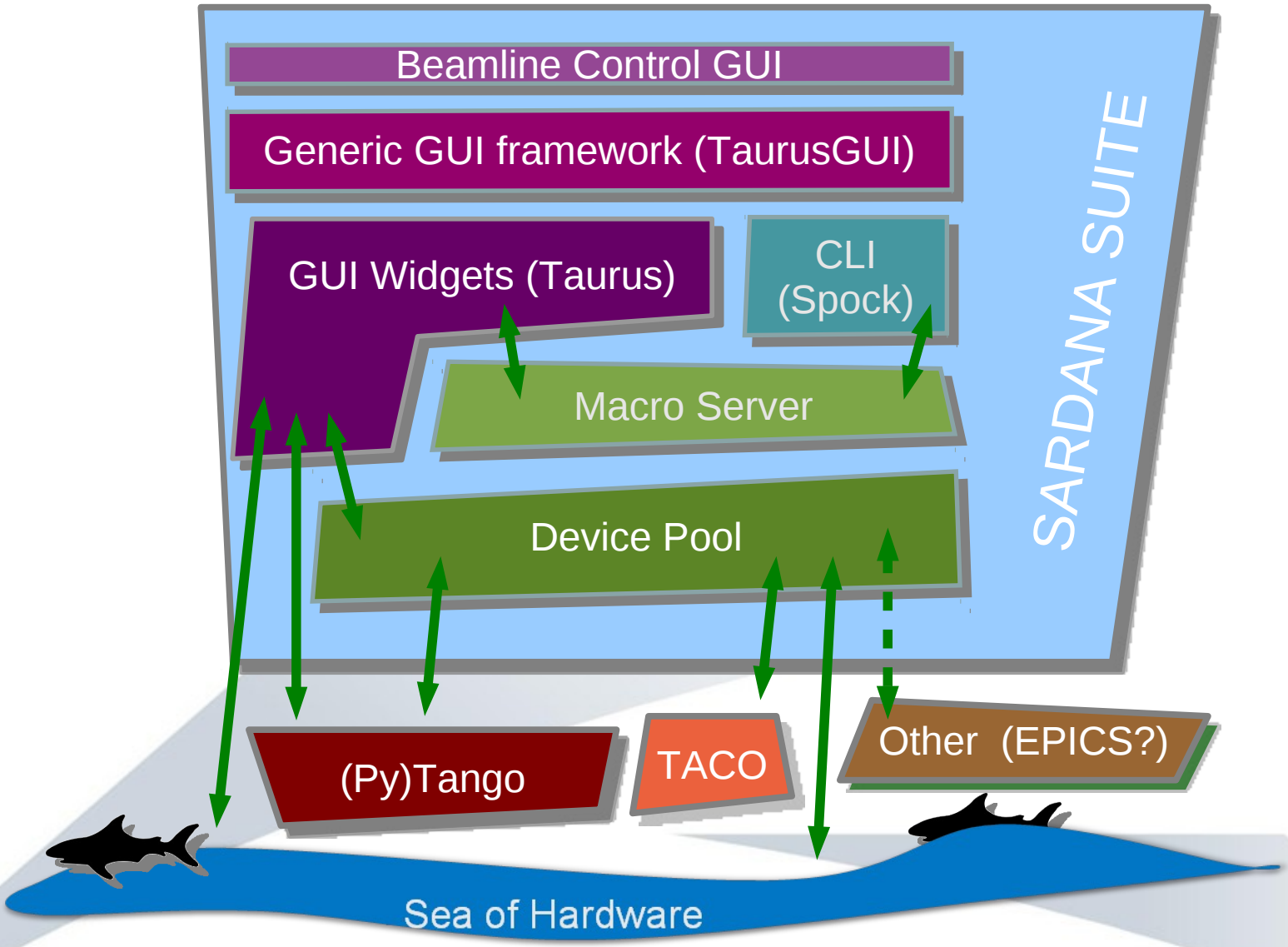
Contributors from:

Online documentation

sardana-controls.org

mailing list: sardana@cells.es

Control system: Sardana

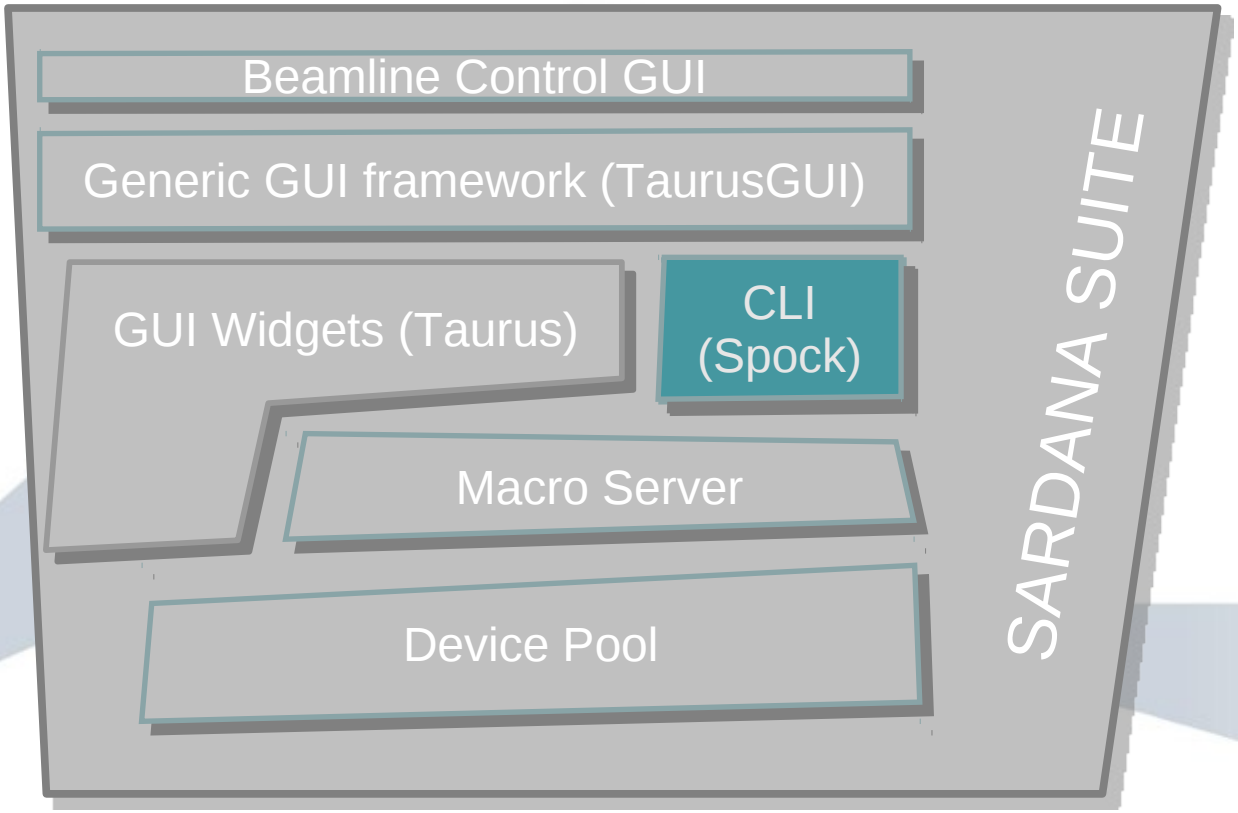


http://sardana-controls.org

sardana@cells.es

CLI: spock

spock: what the user sees of sardana (text mode)



CLI: spock

```
() controls02
File  Edit  View  Bookmarks  Settings  Help
Spock 1.0.0 -- An interactive laboratory application.

help      -> Spock's help system.
object?   -> Details about 'object'. ?object also works, ?? prints more.

door/cpl/1 [1]: lsm
           Name    Type             Controller   Axis
 -------------- ------- --------------------- ------
        m_cpl_1   Motor   dummymotorctrl_cpl_1      1
        m_cpl_2   Motor   dummymotorctrl_cpl_1      2
        m_cpl_3   Motor   dummymotorctrl_cpl_1      3
        m_cpl_4   Motor   dummymotorctrl_cpl_1      4
   m_cpl_ipap_1   Motor             ipap_cpl_1      1

door/cpl/1 [2]: lsmeas
  Active         Name      Timer Experim. channels
 -------- ----------- --------- ---------------------------------------------
    *          cp_mg1   ct_cpl_1 ct_cpl_1, ct_cpl_2, sys/tg_test/1/ulong_image_ro
               cp_mg2   ct_cpl_1 ct_cpl_1, ct_cpl_2
            cp_mg_xas      timer timer, ct_cpl_1, ct_cpl_2

door/cpl/1 [3]: ascan m_cpl_1 0 100 10 .1
Operation will be saved in /home/cpascual/example.h5 (w5)
Scan #166 started at Thu Sep 20 16:37:09 2012. It will take at least 0:00:09.034471
Moving to start positions...
                                                        sys
                                                     tg_test
                                                           1
 #Pt No      dt     m_cpl_1   ct_cpl_1  ct_cpl_2  ulong_image_ro
   0      2.02592      0        0.1       0.2       (251, 251)
   1      2.76325     10        0.1       0.2       (251, 251)
   2      3.49855     20        0.1       0.2       (251, 251)
   3      4.23898     30        0.1       0.2       (251, 251)
   4      4.97804     40        0.1       0.2       (251, 251)
   5      5.71485     50        0.1       0.2       (251, 251)
   6      6.45196     60        0.1       0.2       (251, 251)
   7      7.19135     70        0.1       0.2       (251, 251)
   8       7.9308     80        0.1       0.2       (251, 251)
   9      8.66339     90        0.1       0.2       (251, 251)
   10     9.40609    100        0.1       0.2       (251, 251)
Operation saved in /home/cpascual/example.h5 (w5)
Scan #166 ended at Thu Sep 20 16:37:19 2012, taking 0:00:09.535636. Dead time 88.5% (motion dead time 84.9%)

door/cpl/1 [4]:
() controls02
```

- iPython based
- SPEC-like CLI

	Standard Macros
listing information	lsenv, dumpenv, lsa, lsm, lspm, lsexp, lsior, lsmeas, lsct, ls0d, ls1d, ls2d, lspc, lsctrl, lspc, lsi, lsctrllib, lsa, lsmac, scanhist
moving & acquiring	wa, wm, pwa, pwm, set_lim, set_lm, set_pos, mv, umv, mvr, umvr, ct, uct, settimer
scans step continuous	ascanc, a2scanc, a3scanc, a4scanc, amultiscan, dscanc, d2scanc, d3scanc, d4scanc, dmultiscan, meshc, fscan
other	(u)senv, (u)defmeas, (u)defelem, (u)defctrl, prdef, relmac, relmaclib

CLI: spock (new ipython!)

IPython

File Edit View Kernel Magic Window Help

```
Spock 1.0.0 -- An interactive laboratory application.

help       -> Spock's help system.
object?    -> Details about 'object'. ?object also works,

Welcome to pylab, a matplotlib-based Python environment
module://IPython.zmq.pylab.backend_inline].
For more information, type 'help(pylab)'.

Spock [1]: ascan mot01 0 100 10 0.1
Operation will be saved in /tmp/hh.h5 (w5)
Scan #24 started at Wed Apr 25 14:10:36 2012. It will take a
  #Pt No    mot01      ct01      ct02      ct03
     0       0         0.1       0.2       0.3
     1       10        0.1       0.2       0.3
     2       20        0.1       0.2       0.3
     3       30        0.1       0.2       0.3
     4       40        0.1       0.2       0.3
     5       50        0.1       0.2       0.3
     6       60        0.1       0.2       0.3
     7       70        0.1       0.2       0.3
     8       80        0.1       0.2       0.3
     9       90        0.1       0.2       0.3
    10      100        0.1       0.2       0.3
Operation saved in /tmp/hh.h5 (w5)
Scan #24 ended at Wed Apr 25 14:10:53 2012, taking 0:00:16.8

Spock [2]: wm
```

```
gap02
iceM1
iceM2
mot01
mot02
mot03
mot04
```

QtWidget (embedable)

IPy Untitled0

127.0.0.1:8888/4452407e-cf35-49f9-b1c5-ea80cf3495a8

IP[y]: Notebook Untitled0 Last saved: Apr 25 2:11 PM

File Edit View Insert Cell Kernel Help

Code

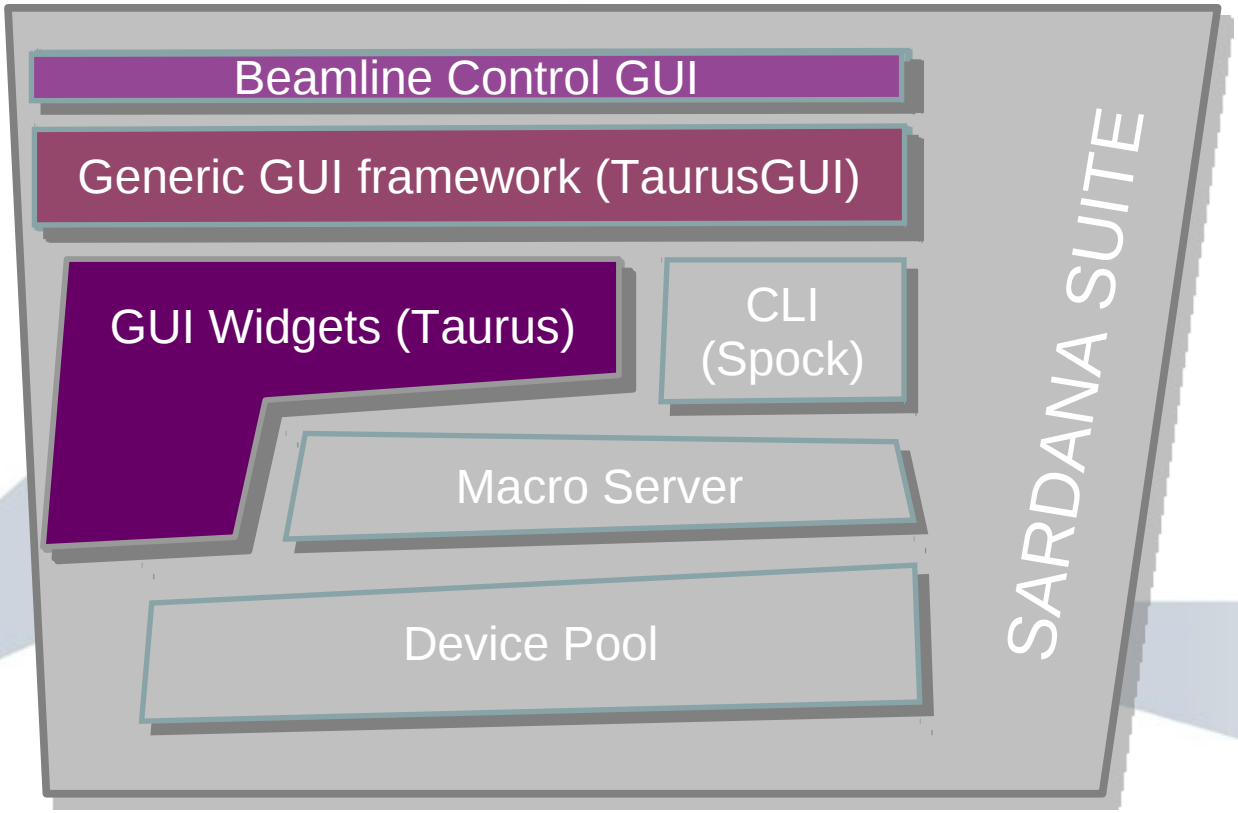
```
In [1]: ascan mot01 0 100 10 0.1
        Operation will be saved in /tmp/hh.h5 (w5)
        Scan #25 started at Wed Apr 25 14:12:24 2012. It will take at least 0:00:01.100000
          #Pt No    mot01      ct01      ct02      ct03
              0       0         0.1       0.2       0.3
              1       10        0.1       0.2       0.3
              2       20        0.1       0.2       0.3
              3       30        0.1       0.2       0.3
              4       40        0.1       0.2       0.3
              5       50        0.1       0.2       0.3
              6       60        0.1       0.2       0.3
              7       70        0.1       0.2       0.3
              8       80        0.1       0.2       0.3
              9       90        0.1       0.2       0.3
             10      100        0.1       0.2       0.3
        Operation saved in /tmp/hh.h5 (w5)
        Scan #25 ended at Wed Apr 25 14:12:41 2012, taking 0:00:16.873967 (dead time was 93.5%)

In [ ]: wm ,
```

```
gap02
iceM1
iceM2
mot01
mot02
mot03
mot04
offset02
sampleoavz
```

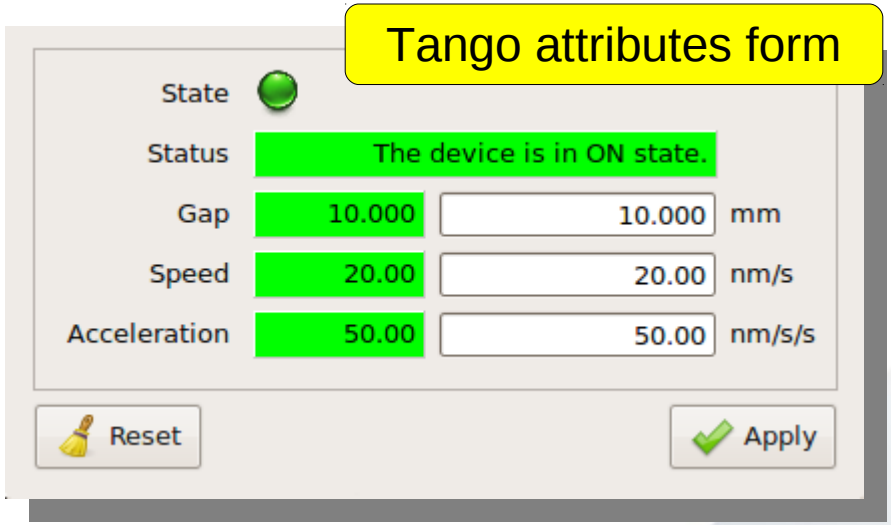
Web-based console!

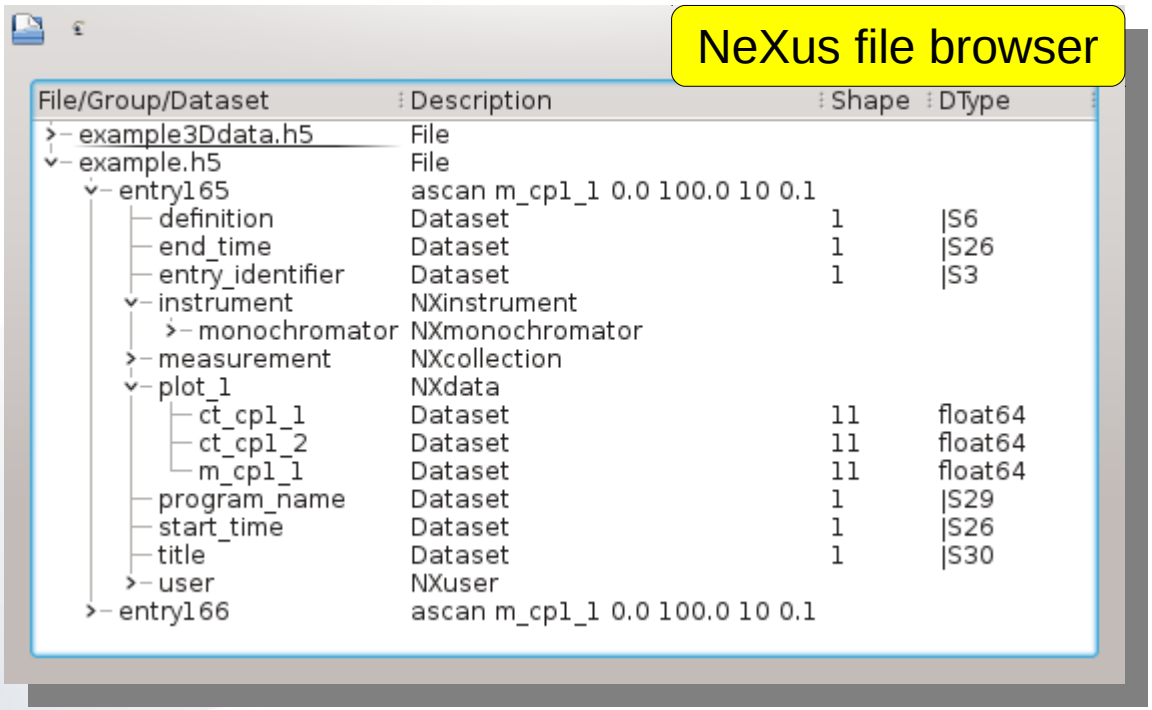
taurusGUI: what the user sees of Sardana (GUI mode)

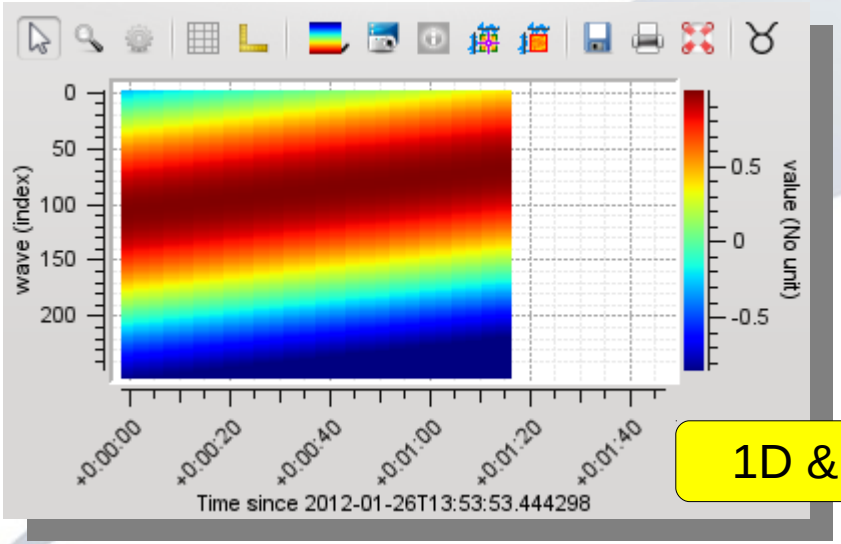


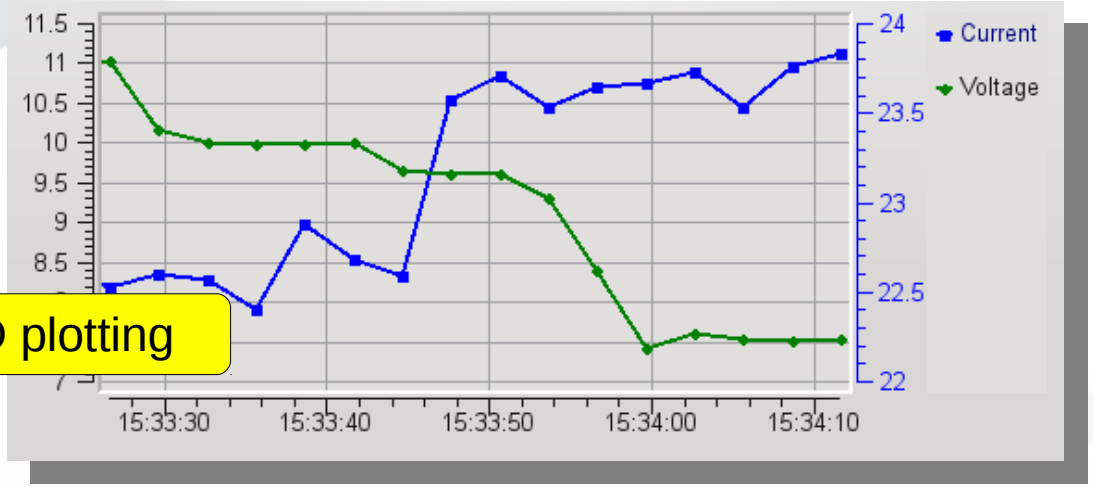
Taurus (generic widgets)

Taurus provides General-purpose widgets









1D & 2D plotting

Widgets (macro-specific)

Experiment configuration

Measurement Group | Snapshot Group | Storage

Active Measurement Group: cp_mg1

Channel	output	Shape	Plot Type	Plot Axes
ct_cp1_1	true	[]	Spectrum	<mov>
ct_cp1_2	true	[]	Spectrum	<mov>
sys/tg_te...	true	[251, 251]	Image	<idx>:<idx>

Reset

And also some specific widgets for dealing with macros

ascan

Parameter	Value
motor	m_cp1_1
start_pos	0.0
final_pos	100.0
nr_interv	10
integ_time	0.1

macro execution

Favourite list | History Viewer

```
mv m_cp1_2 345.0
lsm
ascan m_cp1_1 0.0 100.0 10 0.1
```

72%

ascan m_cp1_1 0.0 100.0 10 0.1

ms/v3/1

- Macro libraries
  - training1
    - where_moveable_2
    - where_moveable
    - snapshot_ccd
    - set_scanid
    - scan_info
    - move_moveable
    - move_energy
    - move2
    - mandelbrot
    - hello_world_2
    - fixed_ascan
    - fft_freq
    - cosine
  - test01
  - submacros
  - standard
  - sequence
  - scans
  - scan
  - parameters
  - mca
  - lists
  - ioregister
  - hooks
  - funcs
  - expert
  - env
  - demo
  - communication

training1.py

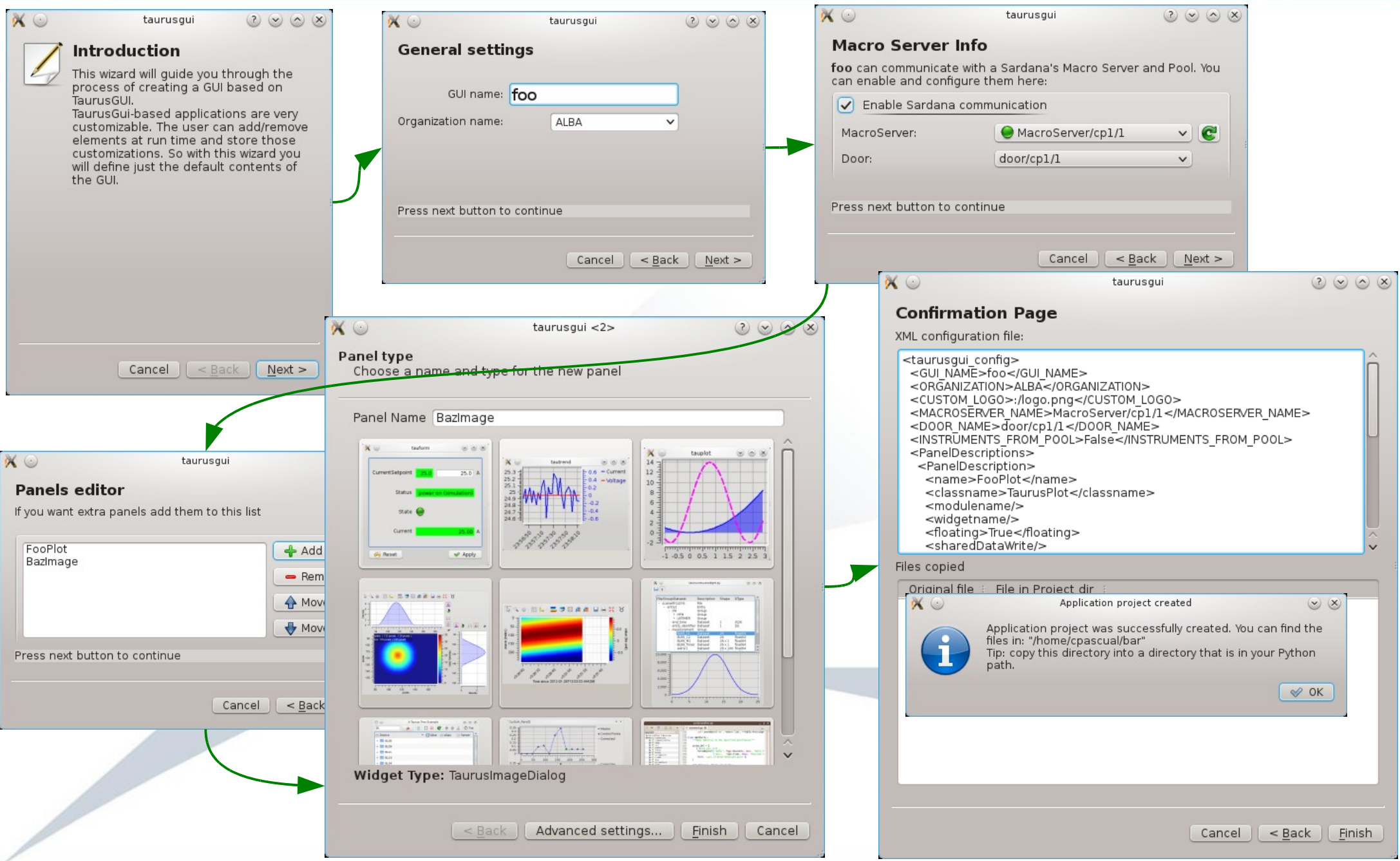
```
71      """This macro changes the current energy"""
72      E = self.getMoveable("Energy")
73      E.move(energy)
74      self.output("Energy is now at %s", E.getPosition())
75
76
77  @macro([ ["ccd",   Type.String, None, "ccd name"],
78           ["fname", Type.String, None, "filename to save"] ]
79  def snapshot_ccd(self, ccd, fname):
80      """Saves the current ccd image to the specified file"""
81      ccd_dev = self.getDevice(ccd)
82      image = PIL.Image.fromarray(ccd_dev.image)
83      image.save(fname)
84
85  @macro([ ["p1", Type.Float, None, "absolute position of mot
86           ["p2", Type.Float, None, "absolute position of mot
87  def move2(self, p1, p2):
88      motion = self.getMotion(["mot01", "mot02"])
89      motion.move([p1, p2])
90
91  @macro([["moveable", Type.Moveable, None, "moveable to get
92  def where_moveable_2(self, moveable):
93      """This macro prints the current moveable position"""
94
95      self.wm(moveable)
96
97
98  @macro([ ["moveable", Type.Moveable, None, "moveable to mov
99           ["position", Type.Float, None, "absolute position"
100 def move_moveable(self, moveable, position):
101     """This macro moves a moveable to the specified positio
102
103     self.mv(moveable, position)
104     self.output("%s is now at %s", moveable.getName(), move
```

Live macro edition

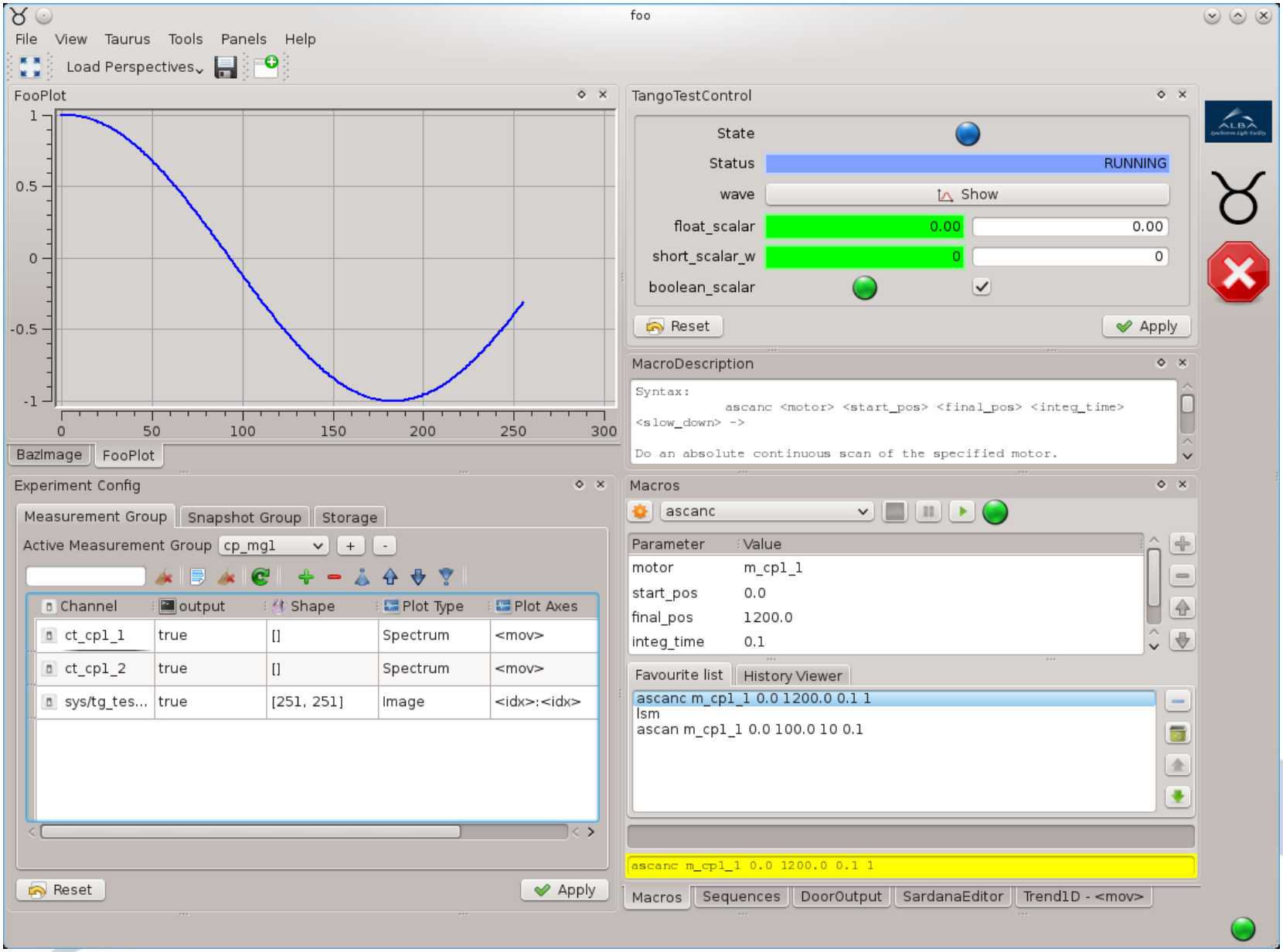
Outline

- training1.py
  - fft_freq
  - hello_world_2
  - where_moveable
  - mandelbrot
  - cosine
  - move_energy
  - snapshot_ccd
  - move2
  - where_moveable_
  - move_moveable
  - fixed_ascan
  - scan_info
  - set_scanid

GUI framework (creating)

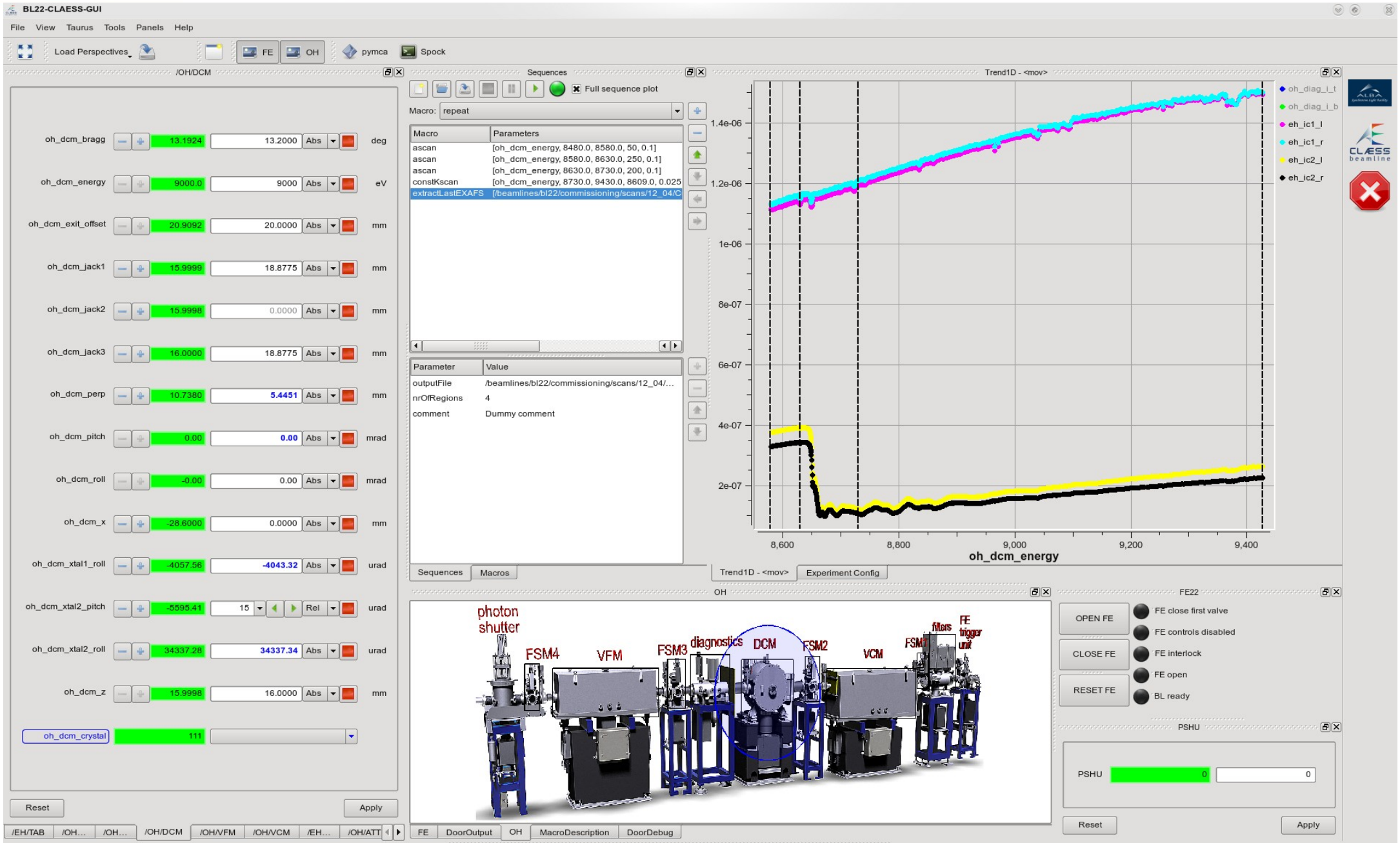


GUI framework (result)

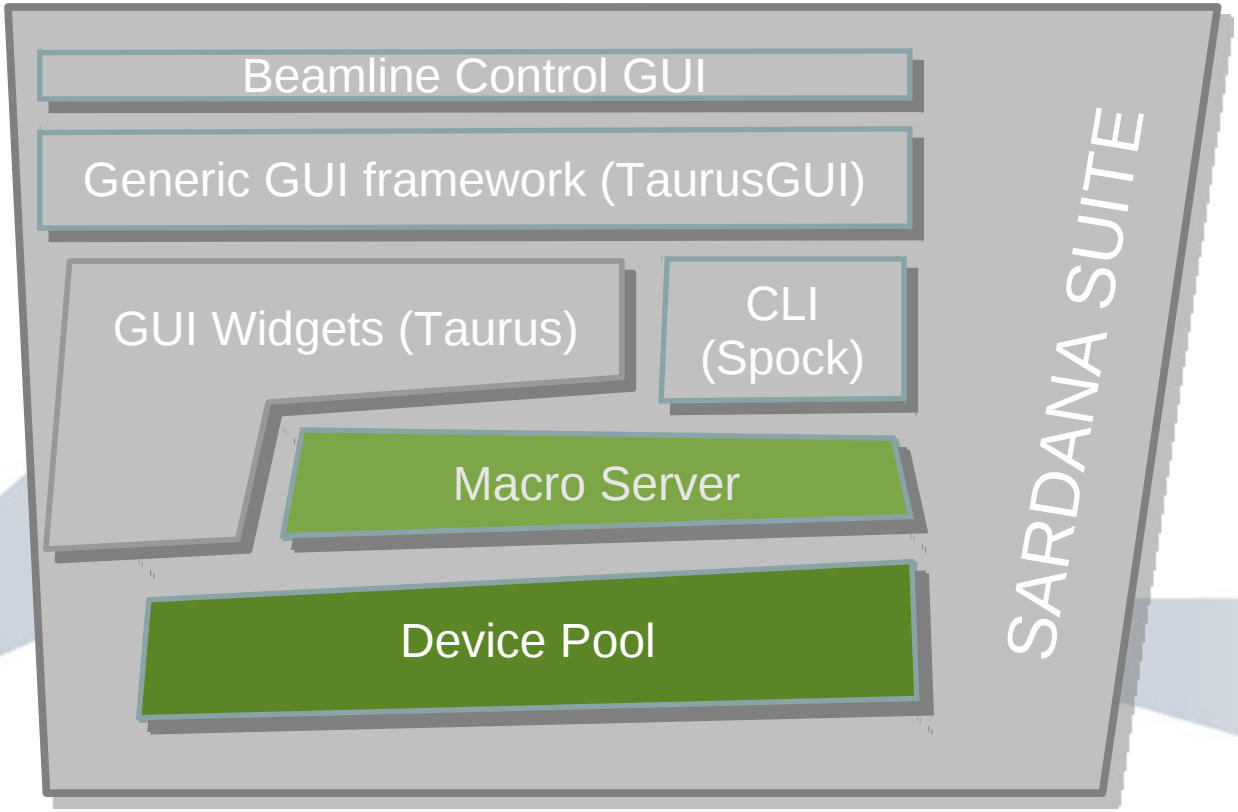


- ✔Dynamic
- ✔User-configurable
- ✔Common Look&Feel
- ✔Based on config file
- ✔Allows perspectives

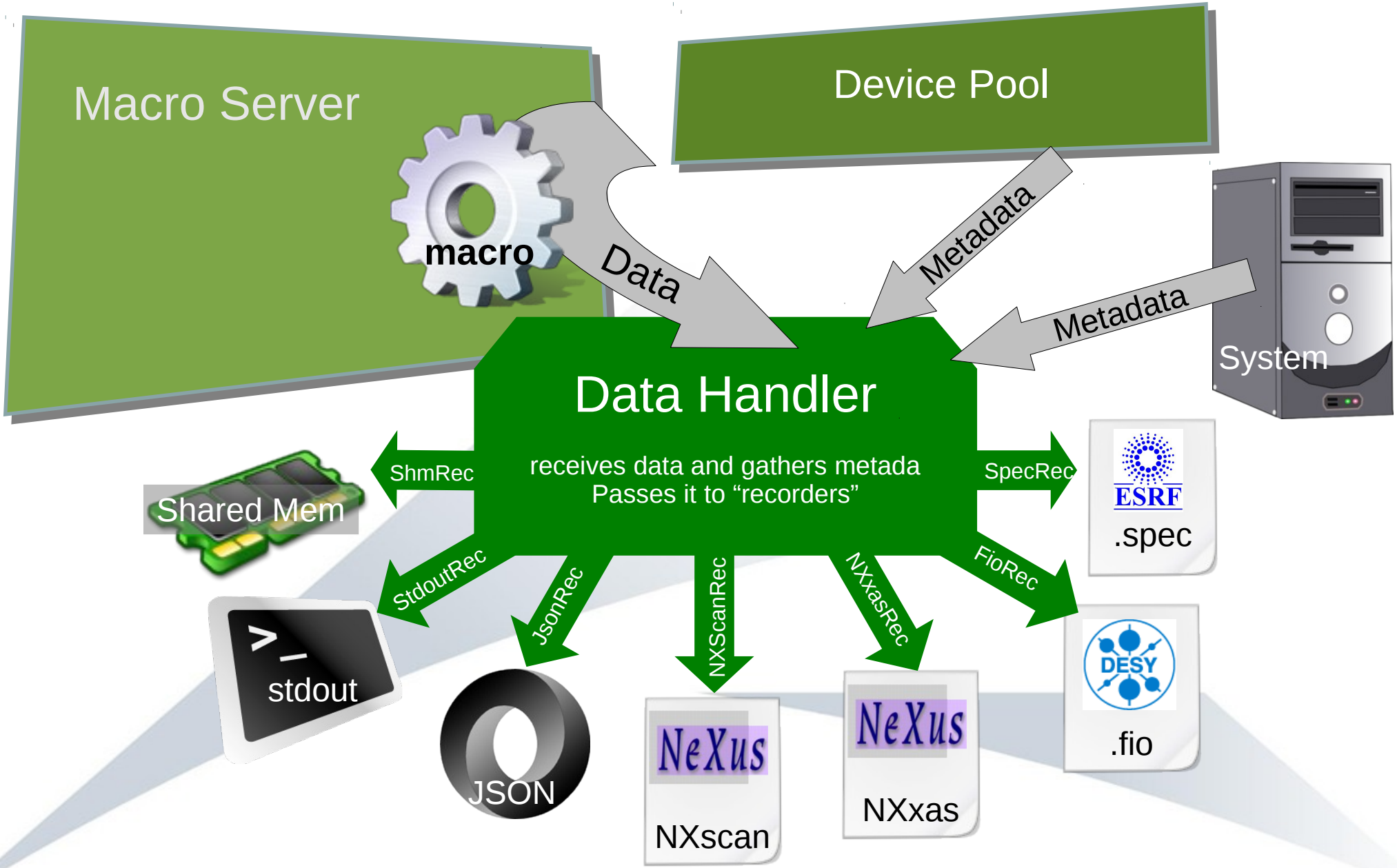
Specific Beamline GUI



MacroServer and Pool: where the data is written (among many other things)



Data handling: recorders



NXscan: tree VS table

Example: a scan where we rotate the sample in 5 steps. In each step we measure the sample temperature and reflectivity during 0.1s .

NXscan "Barebones" NXscan definition.

stdout .spec .fio

entry:NXentry
...
sample:NXsample
angle=[10, 20, 30, 40, 50]
temperature=[44, 55, 66, 77, 88]
instrument:NXinstrument
reflectometer:NXdetector
data=[0.6, 0.7, 0.8, 0.7, 0.6]
timer:NXmonitor
data=[0.1, 0.1, 0.1, 0.1, 0.1]
plot:NXdata
angle--> entry/sample/angle
reflectivity-->entry/instrument/reflectometer/data

#<header containing metadata>...

point#	timer	angle	temperature	reflectivity
0	0.1	10	44	0.6
1	0.1	20	55	0.7
2	0.1	30	66	0.8
3	0.1	40	77	0.7
4	0.1	50	88	0.6

✔Richer metadata
✔Allows multidimensional channels
✘data is scattered along the tree (hard to navigate)

Our NXscan implementation

NXscan "Barebones" NXscan definition.

```
entry:NXentry
    ...
    sample:NXsample
        angle=[10, 20, 30, 40, 50]
        temperature=[44, 55, 66, 77, 88]
    instrument:NXinstrument
        reflectometer:NXdetector
            data=[0.6, 0.7, 0.8, 0.7, 0.6]
    timer:NXmonitor
        data=[0.1, 0.1, 0.1, 0.1, 0.1]
    plot:NXdata
        angle--> entry/sample/angle
        reflectivity-->entry/instrument/reflectometer/data
```

NXscan Sardana's implementation of NXscan

```
entry:NXentry
    ...
    measurement:NXcollection
        angle          =[10, 20, 30, 40, 50]
        temperature    =[44, 55, 66, 77, 88]
        reflectivity   =[0.6, 0.7, 0.8, 0.7, 0.6]
        timer          =[0.1, 0.1, 0.1, 0.1, 0.1]
    sample:NXsample
        angle --> entry/measurement/angle
        temperature --> entry/measurement/temperature
    instrument:NXinstrument
        reflectometer:NXdetector
            data -->entry/measurement/reflectivity
    timer:NXmonitor
        data --> entry/measurement/timer
    plot:NXdata
        angle --> entry/measurement/angle
        reflectivity -->entry/measurement/reflectivity
```

- ✔100% compliant with NXscan application definition
- ✔Data is all in one place (easy to browse&convert)
- ✔Easy to dump scalar channels to a table
- ✔More natural for SPEC users

Other NeXus definitions

- **Goal:** allow specific macros to write NeXus files with application definitions other than NXscan

- **Proof of concept: *xas_acq*** custom macro which uses NXxas recorder:

```
from sardana.macroserver import Macro
class xas_acq(Macro):
    """
    Perform a X-ray absorption scan experiment. Data is stored in a NXxas-compliant file.
    """
    hints = { 'FileRecorder':'NXxas_FileRecorder', 'scan' : 'xas_acq' }

    param_def = [['start',      Type.Float,    None, 'start energy in keV'],
                 ['final',      Type.Float,    None, 'final energy in keV'],
                 ['nr_interv',  Type.Integer,  None, 'Number of energy intervals'],
                 ['integ_time', Type.Float,    None, 'Integration time in s']]


    def prepare(self, start, final, nr_interv, integ_time, **opts):
        ...

    def run(self,*args):
        ...
```

Our NXxas implementation

NXxas

```
root:NXroot

  @HDF5_Version = 1.8.1
  @NeXus_version = 4.3.0
  @file_name = /home/cpascual/test_xas2.h5
  @file_time = 2012-09-21T18:50:33+01:00

  entry192:NXentry
    title = xas_acq 1.0 4.0 5 0.1
    definition = NXxas
    start_time = 2012-09-21T18:50:33.177349
    end_time = 2012-09-21T18:50:35.042047
    epoch = 1348246233.0
    entry_identifier = 192
    program_name = sardana (NXxas_FileRecorder)

    measurement:NXcollection
      Pt_No = [0 1 2 3 4 5]
      dt = [ 0.39  0.66  0.92  1.18  1.45  1.73]
      ct_cp1_1 = [ 0.1  0.1  0.1  0.1  0.1  0.1]
      ct_cp1_2 = [ 0.2  0.2  0.2  0.2  0.2  0.2]
      m_cp1_1 = [ 1.   1.6  2.2  2.8  3.4  4. ]
      timer = [ 0.1  0.1  0.1  0.1  0.1  0.1]

    data:NXdata
      absorbed_beam --> /entry192/measurement/ct_cp1_2
      energy --> /entry192/measurement/m_cp1_1

    sample:NXsample
      name = Unknown

    user:NXuser
      name = cpascual
                                                  ...
```

```
...
    monitor:NXmonitor
      data --> /entry192/measurement/timer
      mode = timer
      preset = 0.1

    instrument:NXinstrument
      absorbed_beam:NXdetector
        data --> /entry192/measurement/ct_cp1_2
      incoming_beam:NXdetector
        data --> /entry192/measurement/ct_cp1_1
      monochromator:NXmonochromator
        energy --> /entry192/measurement/m_cp1_1
      source:NXsource
        name = ALBA
        probe = x-ray
        type = synchrotron
```

- ✔100% compliant with NXxas application definition
- ✔All data is in “measurement”, with internal sardana names
- ✔Named links provide translation
- ✔Compatible with NXscan
- ✔NXxas recorder uses PyNXtree API

Pending issues

- **We haven't had yet time to:**
  - Write more NX recorders and macros (NXtomo, NXfluo, ...)
  - Write specialised “Experiment description” widgets (now using generic one)
  - Do better handling of metadata (now sources are disperse and ad-hoc)
  - Writing NeXus in parallel...
  - ...or at least integrating externally written hdf5 files into a main file

- **We already tried but failed:**
  - Deleting (or hiding) data in NeXus files (needed for tagging data)
  - Creating named links with the new PyNXtree API
  - Writing arrays of strings in a reasonable way (static allocation for strings is ugly!)
  - Convince BL scientists that NeXus=Good and SPEC=Trouble ;-)

The culprits: Alba BL controls

Tiago Coutinho
PyTango (all)
MServer and Pool (all)
Taurus (a lot)
Scan framework
The universe in 6 days?

Carlos Pascual
Taurus (a bit)
NeXus support
Scan framework
Pestering Tiago (a lot)

David Fernández
General guidance
while trying not
to be bossy

Guifré Cuní
Pool widwets,
Pool elements
Scan framework
BL11 & BL13 GUIs

Zbigniew Reszela
MacroExecutor &Co
Pool Elements
BL22 & BL04 GUIs

Jörg Klora
Chief salesman

Antonio Milán
Frontend Widget
Pool elements
BL09 & BL24 GUIs

Sergi Rubio
Vacuum widgets
Alarm widgets
Archiving integration

Jairo Moldes
BL29 GUI
Pool elements
Bug spotting

Sergi Blanch
Taurus LIMA support
CCDs

Gabriel Jover
LIMA-pool integration