Sardana & Taurus status

by:

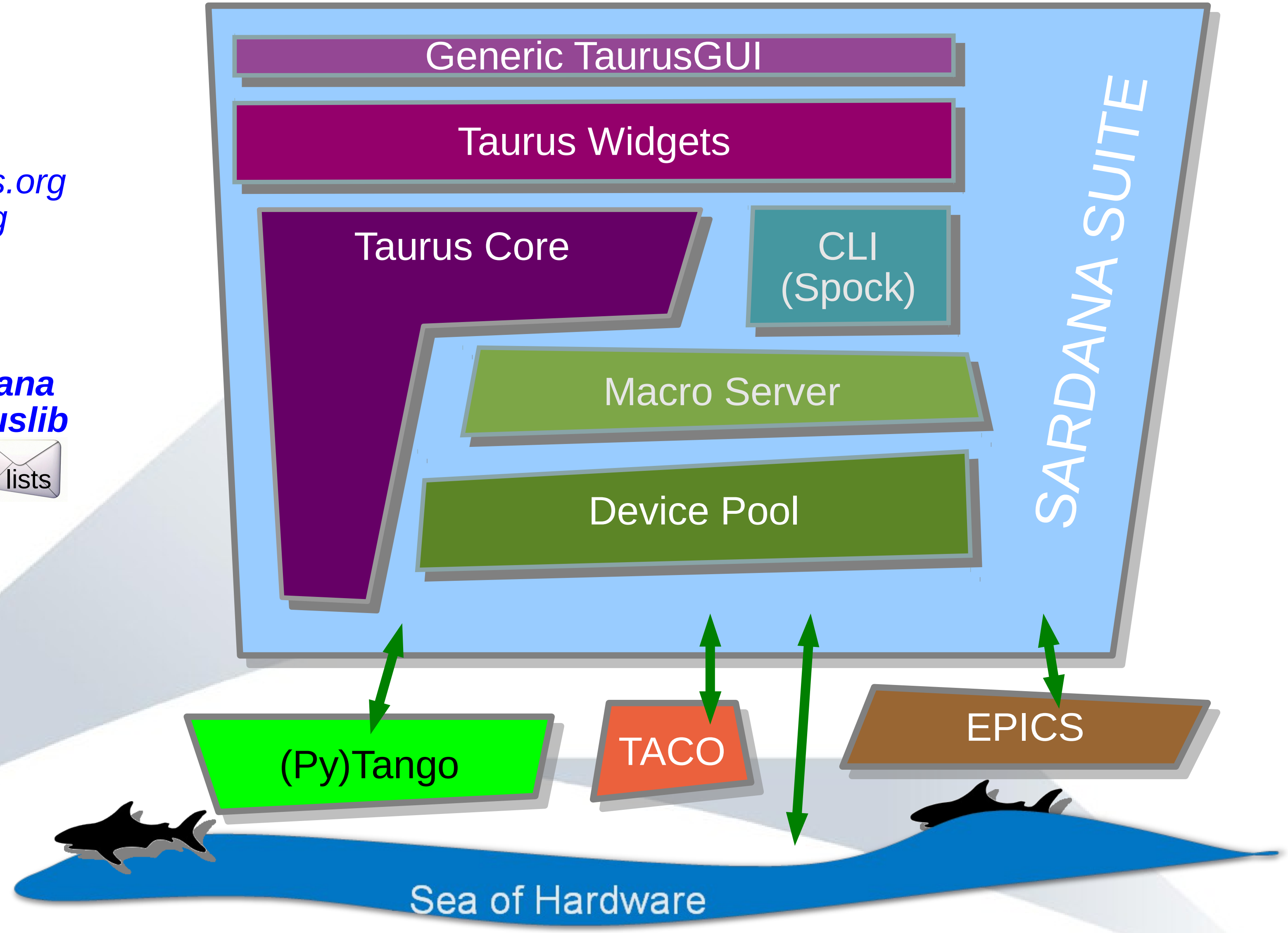
Carlos Pascual-Izarra
&
Zbigniew Reszela

(on behalf of the Sardana Community)

What are Sardana & Taurus?

Docs http://sardana-controls.org
http://taurus-scada.org

http://sf.net/p/**sardana**
http://sf.net/p/**tauruslib**
git [wiki] lists



Sardana, Taurus and Tango

2011-08

- Taurus is born (from Tau)
- Everything hosted in tango-cs.sf.net
- Sdn partially C++

Jul13 *release*

2013-08

- Sdn moves to its own project
- Sdn starts importing Taurus
- Sdn becomes 100% Python

Jan14 *release*

2014-03

- Sdn comunity is born
- Sdn & Taurus migrate to Git
- Taurus moves to sardana.sf.net

(now)

2014-07

- Sdn becomes a plugin for Taurus (SEP10, step1)
- tauruslib.sf.net project is created

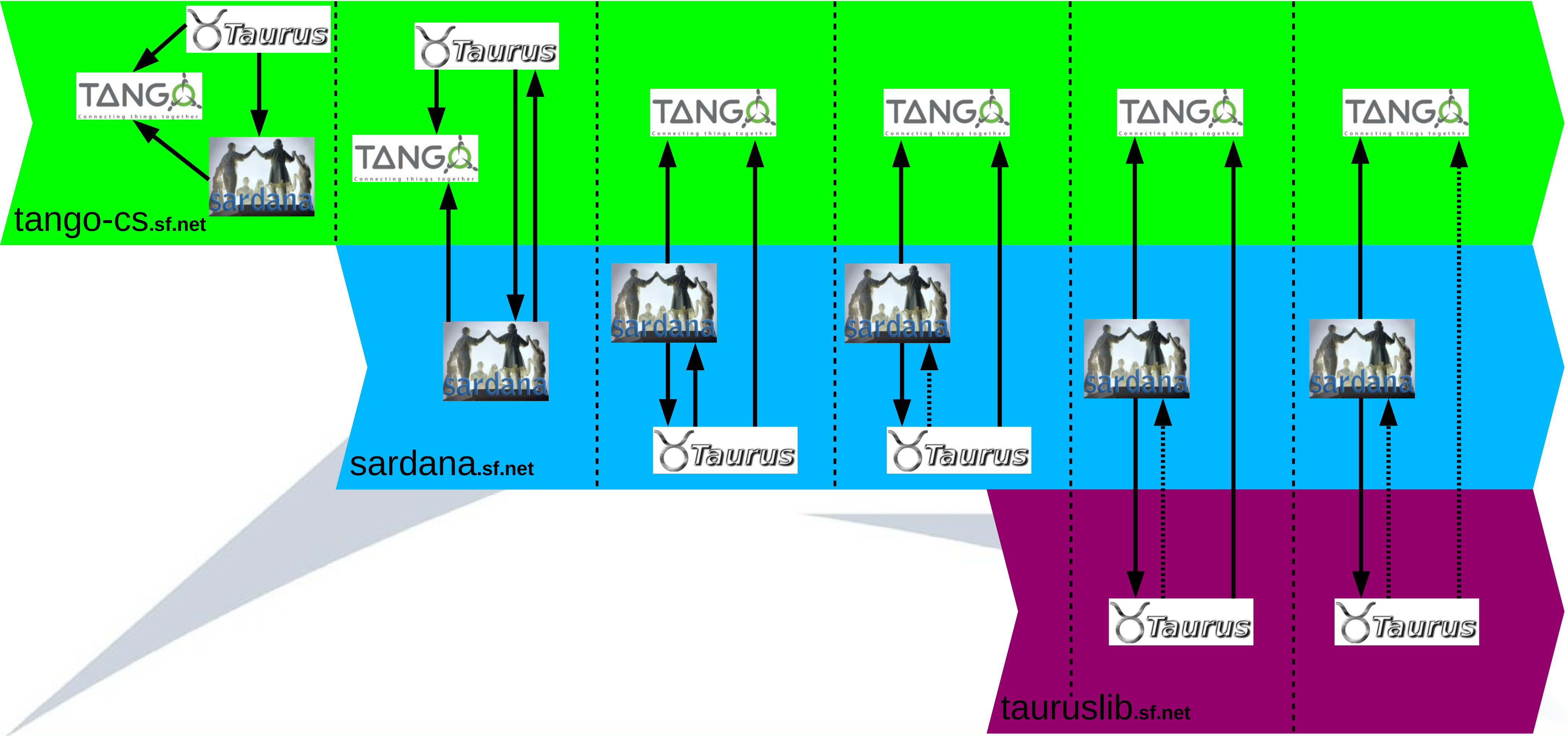
Jul14 *release*

2014-0?

- Taurus moves to tauruslib.sf.net (SEP10, step2)

Jan15 *release*

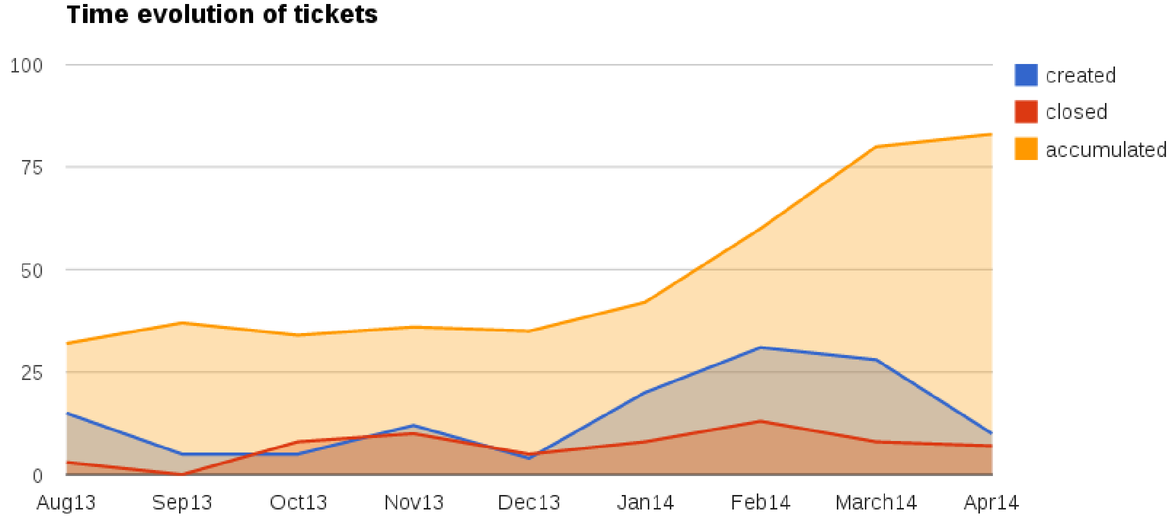
- Tango becomes optional for Taurus (SEP3 + SEPx)
- ...

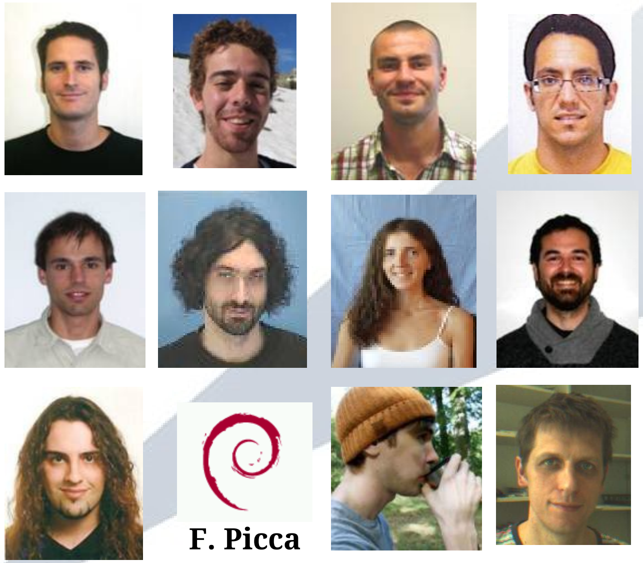


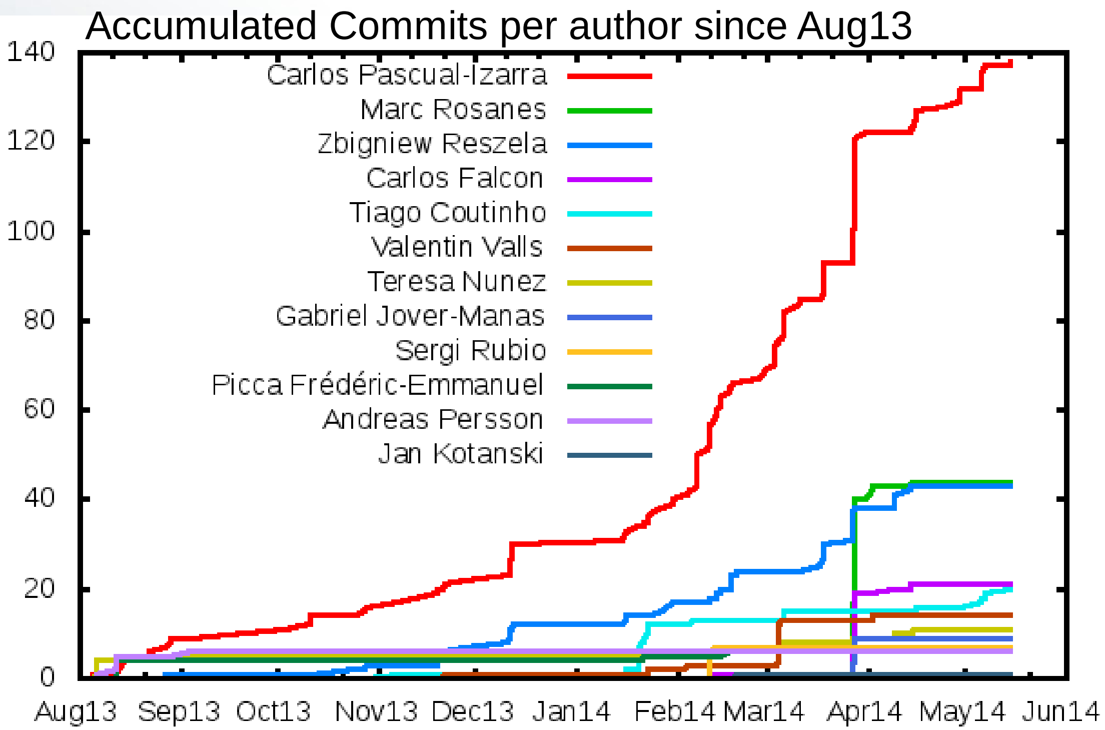
Sardana Community stats

The last 9 months in Numbers:

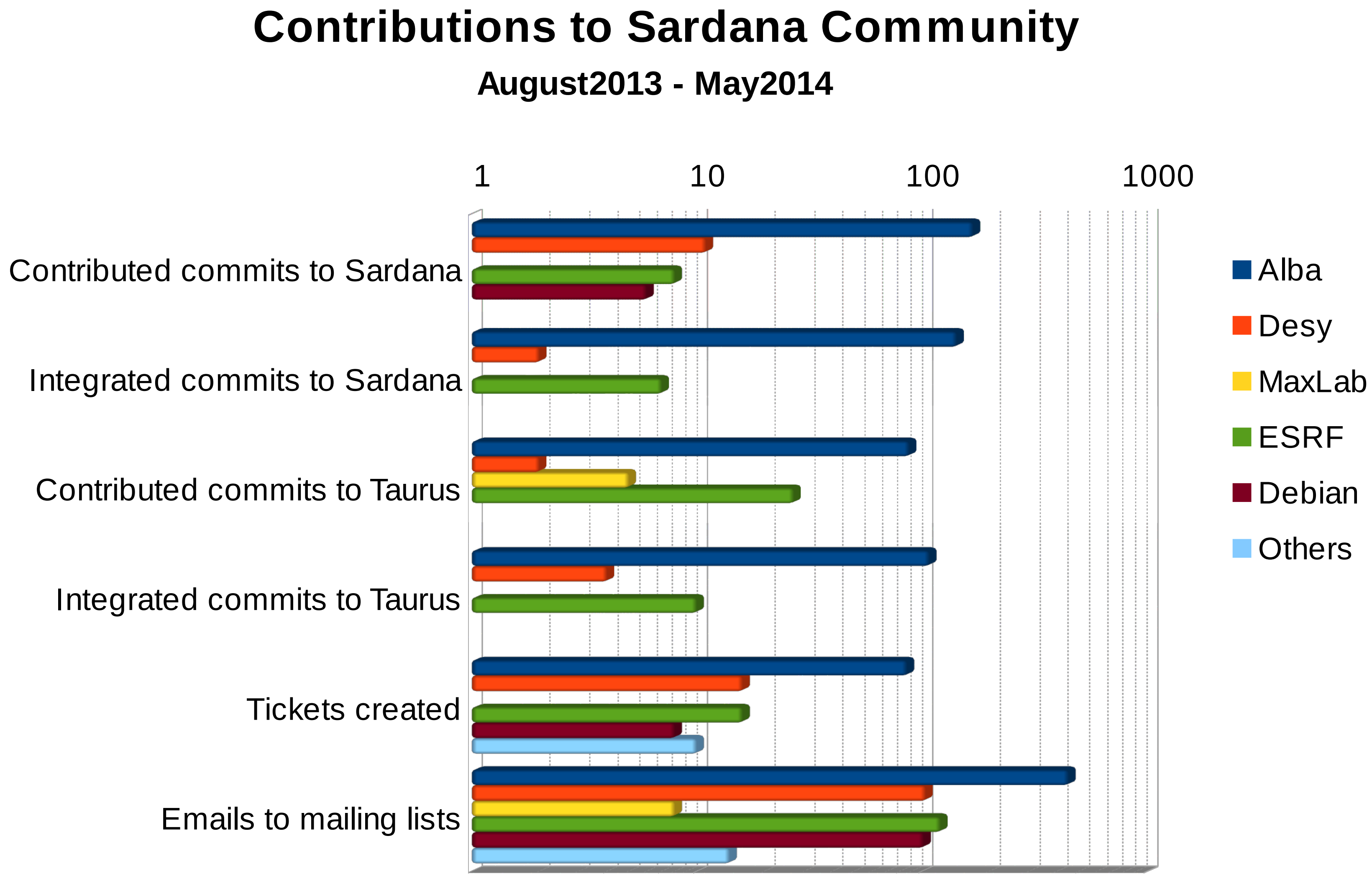
- **1 release** (+1 next July)
- **12 SEPs** (5 already finished)
- **317 commits** (+ more in branches)
- **137 tickets** (58 solved)
- **804 emails** (~220 threads)







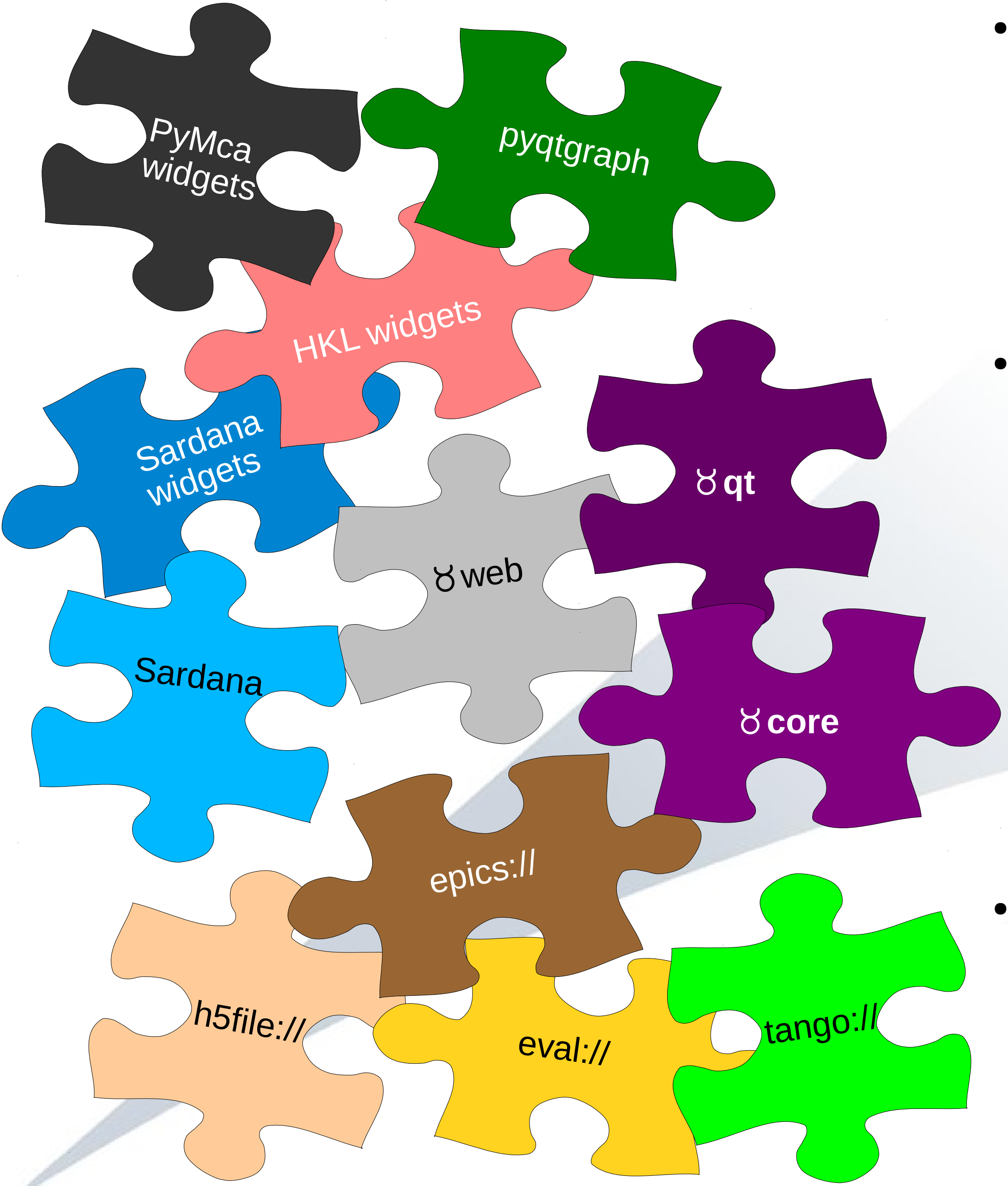
Sardana Community stats



Next in Taurus

Plug-in system	💡	💬	🛠	
Use of Pint Quantities	💡	💬	🛠	
Multi-models (adapter pattern)	💡	💬	🛠	
Replace Qwt for plots	💡	💬		
Allow external logging (SEP8)	💡	💬	🛠	🔍
Compact read-write widgets (SEP9)	💡	💬	🛠	🔍
Direct Load of ui files (SEP11)	💡	💬	🛠	🔍
Use of standard Enum (SEP12)	💡	💬	🛠	
Use ReadThe Docs for documentation	💡	💬	🛠	
Direct registering of Icons (avoid resource files)	💡	💬	🛠	
Merge TaurusConfiguration into TaurusAttribute	💡	💬		
Create the h5file:// scheme	💡			
New style signals (drop support of PyQt<4.5)	💡			
Support Python3	💡			
Support PySide and Qt5	💡	💬	🛠	
Introduce QML widgets	💡			
Refactor Taurus Polling Loop	💡			
Generic support for archiving values	💡			

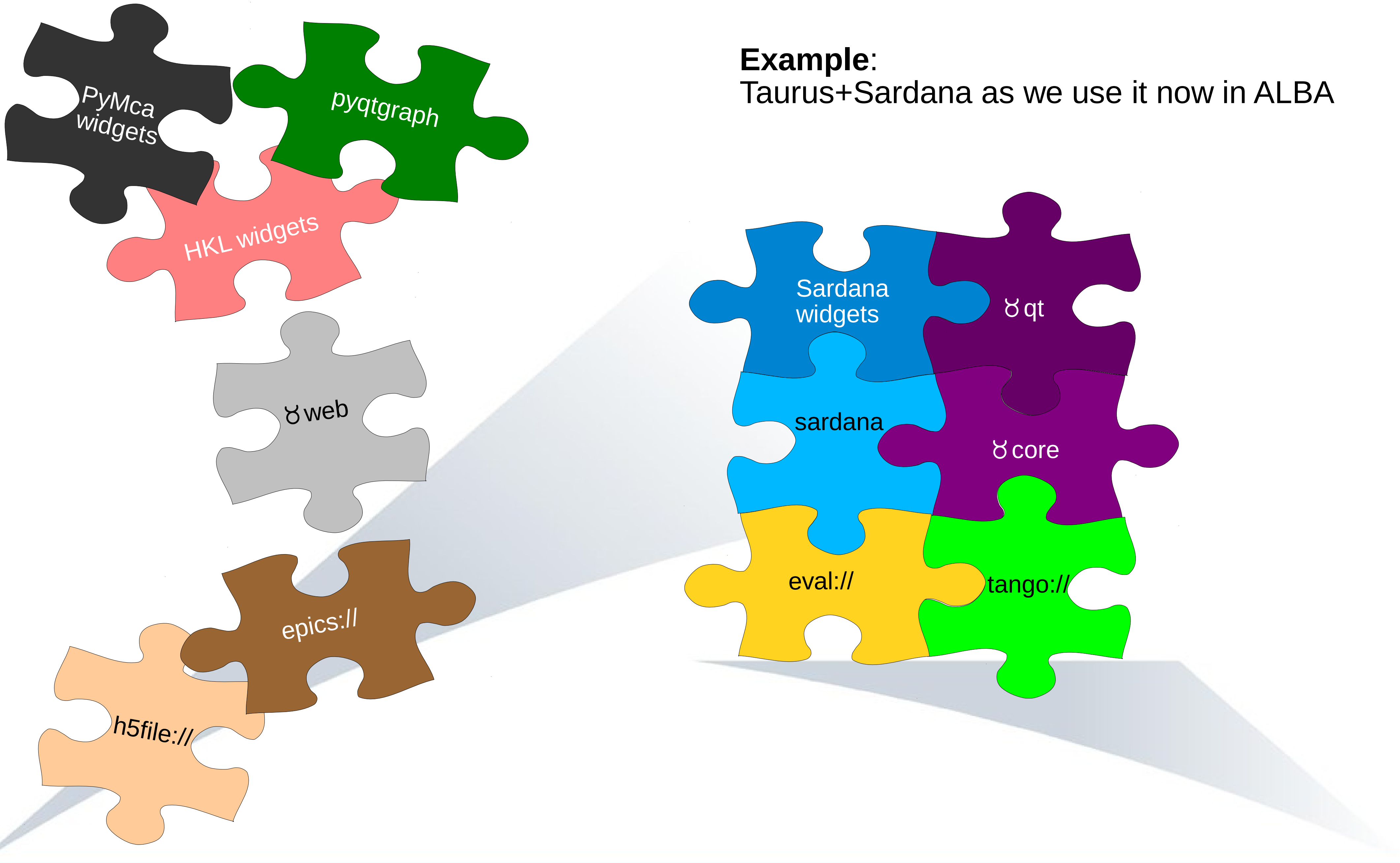
Next in Taurus: plugins



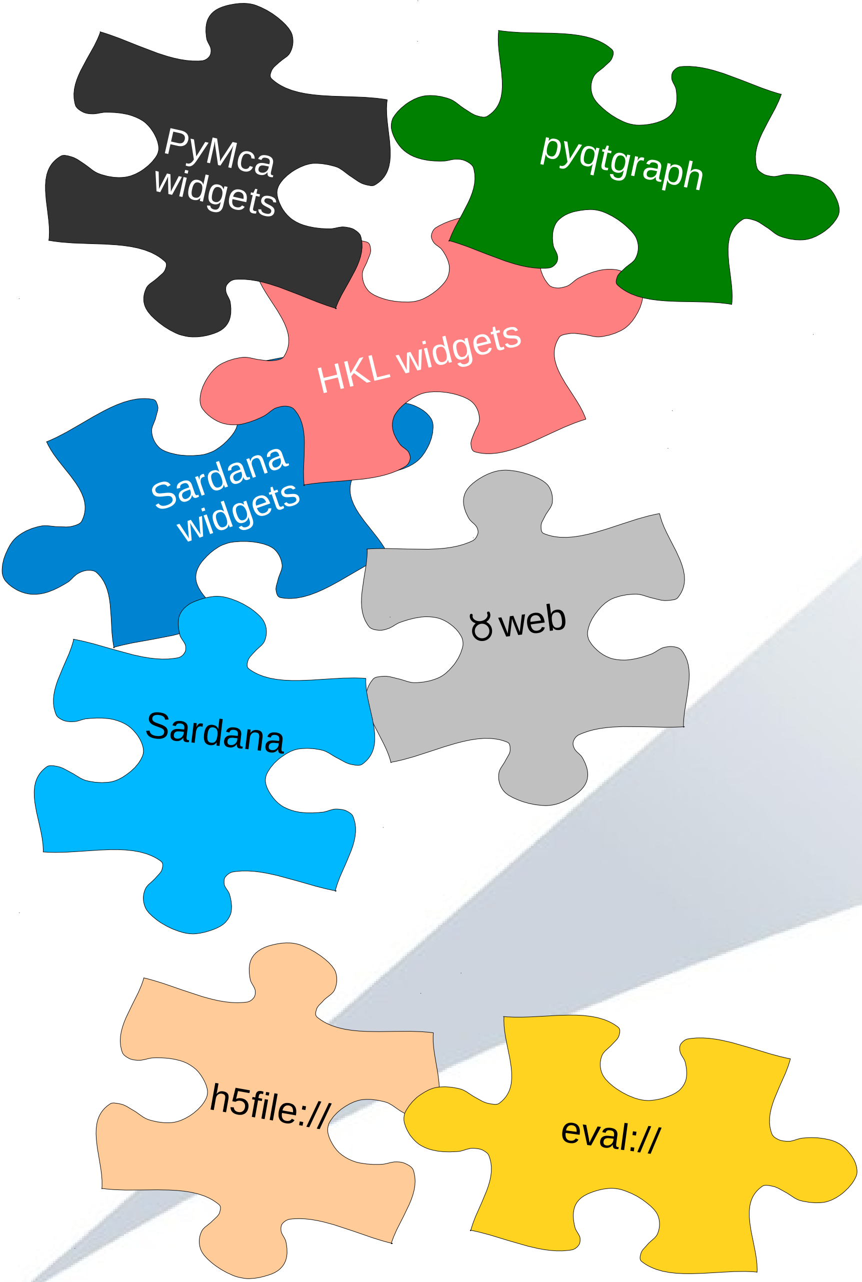
- **Plugins will make Taurus...**
  - Light: most dependencies optional
  - Usable as a library for data analysis GUIs (not connected to control system at all)
  - Extendable for user specific needs
- **What could be a Taurus plugin?:**
  - Schemes (data sources)
  - Widgets: user-created or domain-specific
  - Taurus Core extensions (e.g., Motor device especialization of a TaurusDevice)
  - TaurusGui's “New Panel” catalog
  - Component of certain widgets (e.g., macroserver configuration in “new GUI” wizard)
- **What could be a Sardana plugin?:**
  - Macros
  - Controllers
  - DataRecorders
  - Custom parameter editors for MacroExecutor

Next in Taurus: plugins

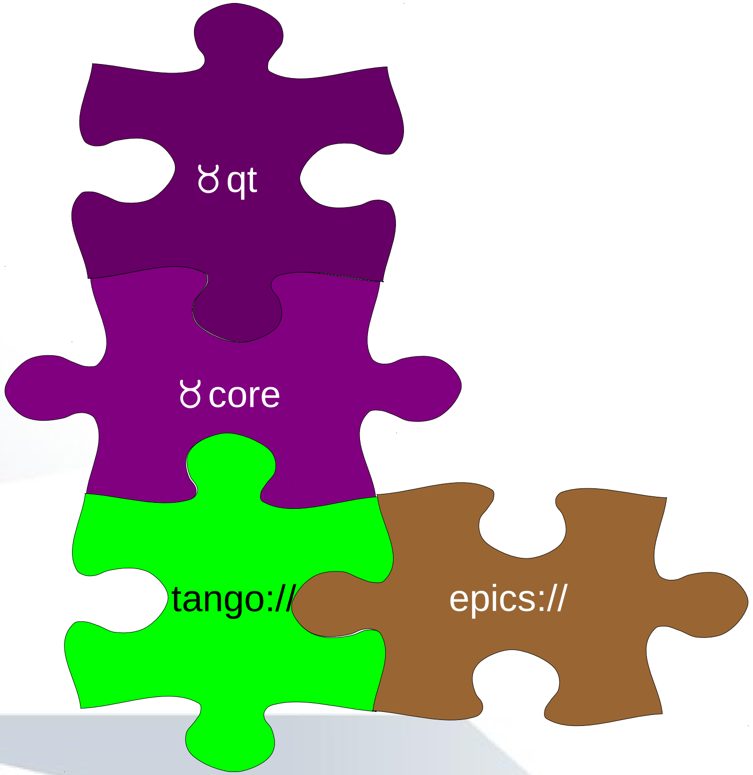
Example:
Taurus+Sardana as we use it now in ALBA



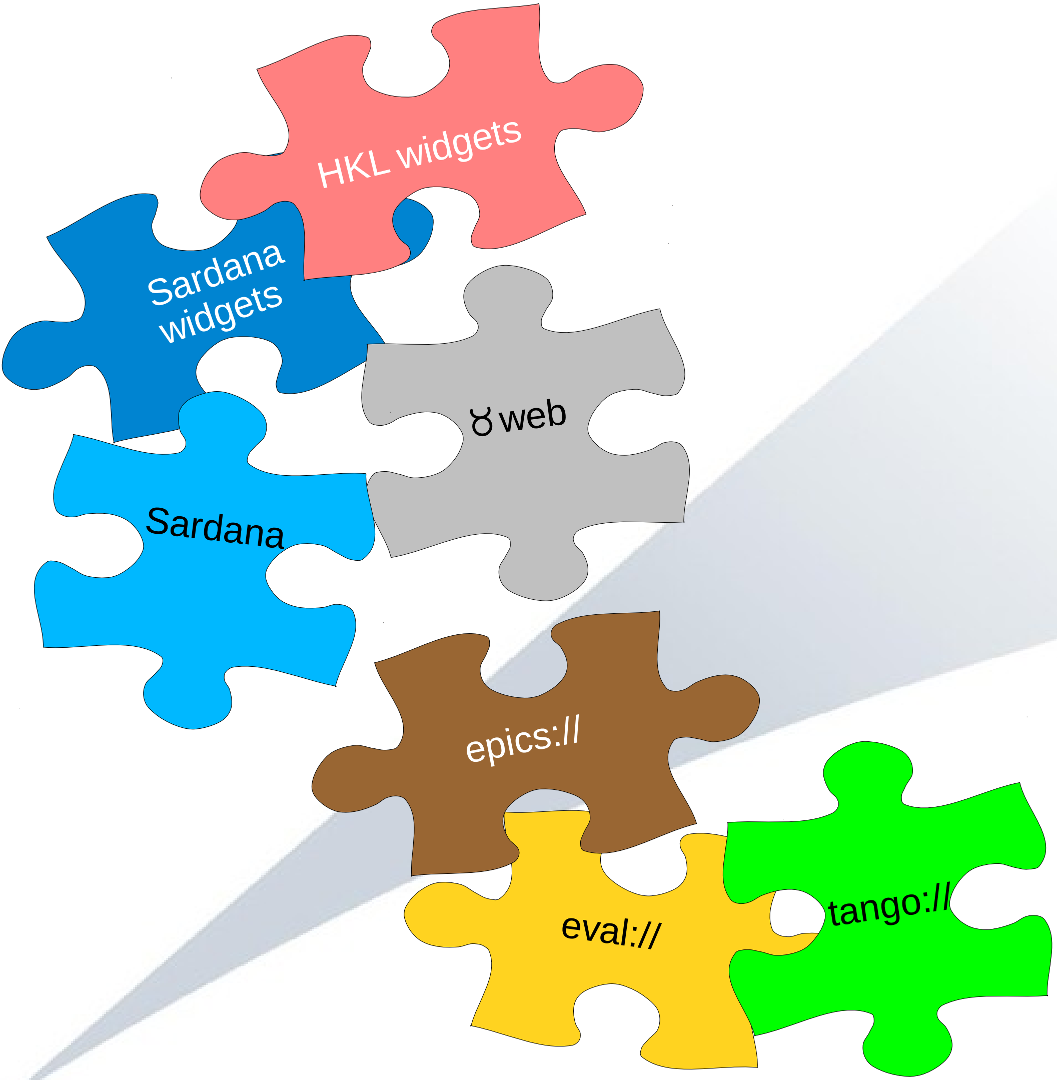
Next in Taurus: plugins



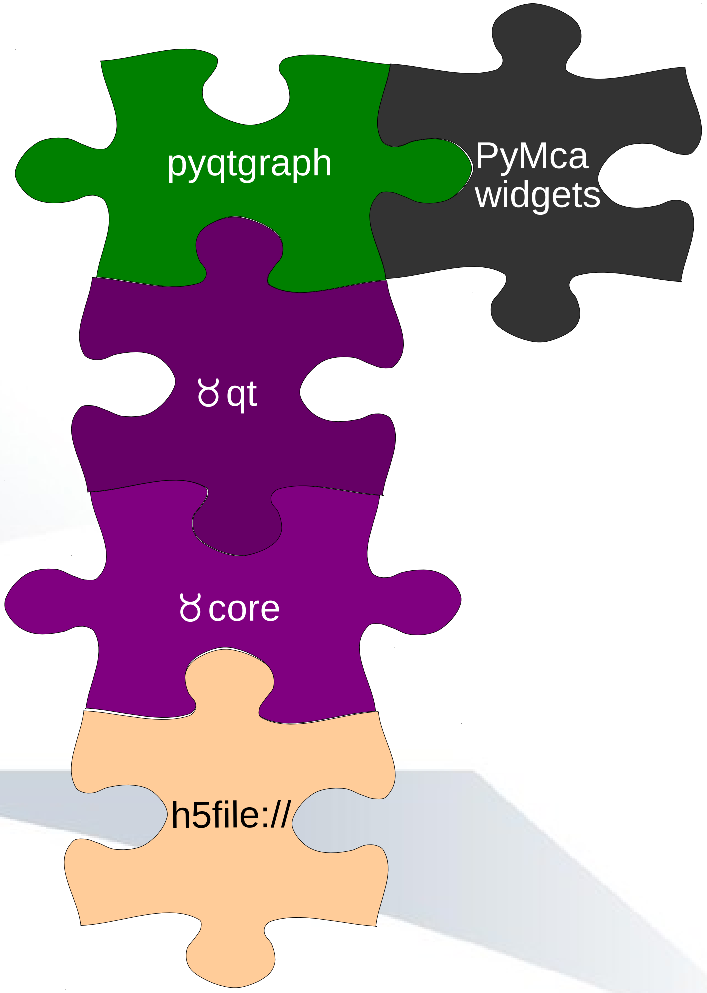
Example:
Controlling a mixed Tango+EPICS environment



Next in Taurus: plugins

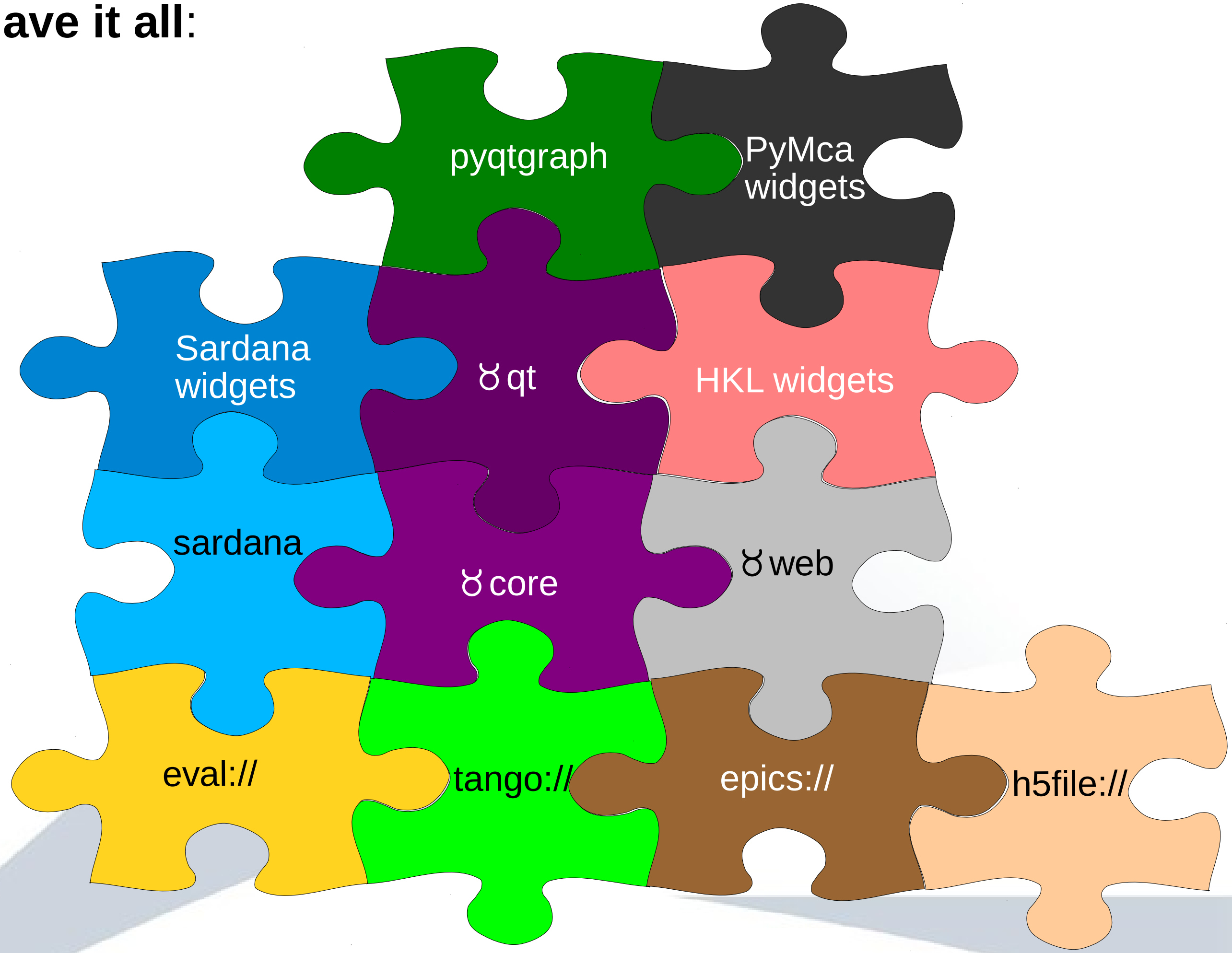


Example:
Taurus for Data Analysis (no control system)



Next in Taurus: plugins

... or have it all:



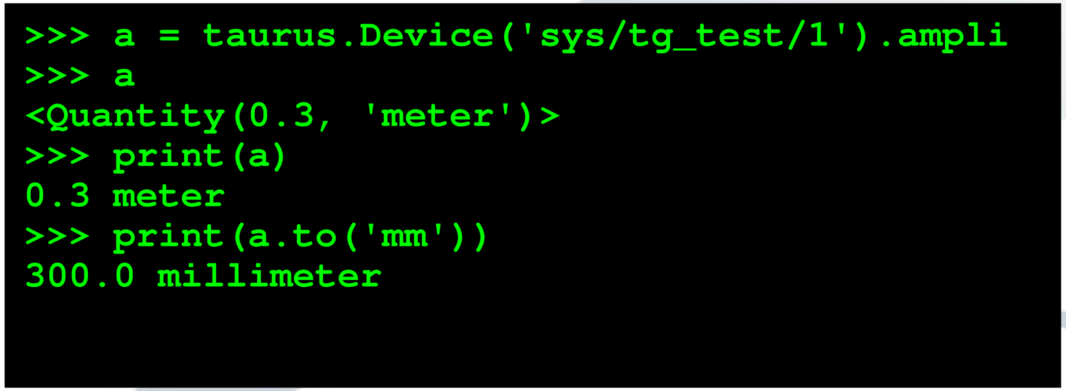
See docs about plugin patterns in:
http://stevedore.readthedocs.org

- More details in Tiago's talk in the Sardana Meeting

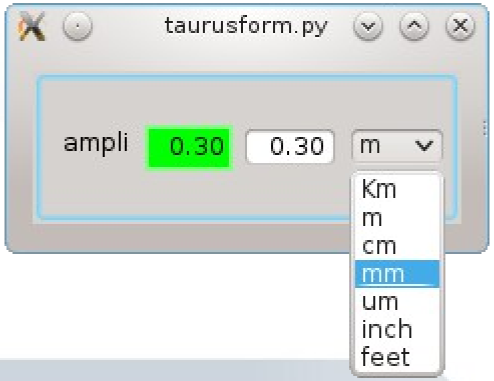
Next in Taurus: quantities

- All values in Taurus (and their limits, etc.) will be **Pint** “Quantities”
- Quantities can be converted & used in operations
- Compatible with numpy arrays
- No extra dependency (single file module, bundled in Taurus)
- Each scheme is responsible of encoding/decoding the units to Pint
- Widgets may use it to offer user-friendly input/output
- Proof of concept ready (to be included in SEP3)

http://pint.readthedocs.org



```
>>> a = taurus.Device('sys/tg_test/1').ampli
>>> a
<Quantity(0.3, 'meter')>
>>> print(a)
0.3 meter
>>> print(a.to('mm'))
300.0 millimeter
```



- More details in Tiago's talk in the Sardana Meeting

Next in Taurus: multimodels

- **Current Situation:**
  - widgets inherit from TaurusBaseComponent
  - base classes assume connection to **just 1 model**
  - Other cases require particular solutions

```
# standard widgets connect to 1 model
e = TaurusValueLineEdit()
e.model = 'a/b/c/d'

# but some parse the name and
# connect to other things!
m = PoolMotorTV()
m.model = 'a/b/c' # but it connects to:
                  #   'a/b/c/pos'
                  #   'a/b/c/state'

# and some widgets receive a list
# to manage it on their own!
f = TaurusForm()
f.model = ['a/b/c/d', 'e/f/g/h']
```

- **Comming soon**: use the **adapter pattern**
  - widgets implemented as a pure Qt
  - model-management methods are inserted by **adapter decorators**
  - Multiple models supported with homogeneous API

```
@modelable('positionModel')
@modelable('stateModel')
class NewMotor(QWidget):
    ...

m = NewMotor()
m.stateModel = 'a/b/c/state'
m.positionModel = 'a/b/c/pos'
```

```
@list_modelable
class NewPlot(PyQtGraph.PlotWidget):
    ...

p = NewPlot()
m.model = ['a/b/c/d', 'e/f/g/h']
```

- More details in Tiago's talk in the Sardana Meeting

Next in Taurus: bye qwt!

Taurus uses PyQwt for plotting, but...

- PyQwt is **no longer maintained**
- 3D support may be required soon

•**Vispy** (http://vispy.org)

- ✓ OpenGL based.
- ✓ Focus on large data sets and 3D visualization
- ✓ Collaboration of several developers
- ✗ Not yet ready

•**matplotlib**:(http://matplotlib.org)

- ✓ Widely used. Large community
- ✓ Well tested, documented and stable API
- ✗ Limited 3D
- ✗ Focused on static plots
- ✗ Qt backend not a priority (may no longer be true)

•**PyQtGraph**: (http://pyqtgraph.org)

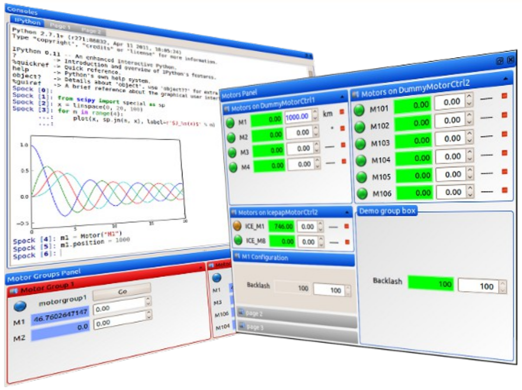
- ✓ Pure python module (no extra dependencies)
- ✓ Fast: numpy, QGraphicsView, OpenGL
- ✓ Will integrate Vispy (when ready)
- ✗ Single main developer

} **PyMca.**PyMcaGraph backends

GRAHAM CHAPMAN · JOHN CLEESE · TERRY GILLIAM · ERIC IDLE · TERRY JONES · MICHAEL PALIN
MONTY PYTHON'S
AND NOW FOR SOMETHING COMPLETELY DIFFERENT
DVD
THE BEST OF MONTY PYTHON'S FLYING CIRCUS
12

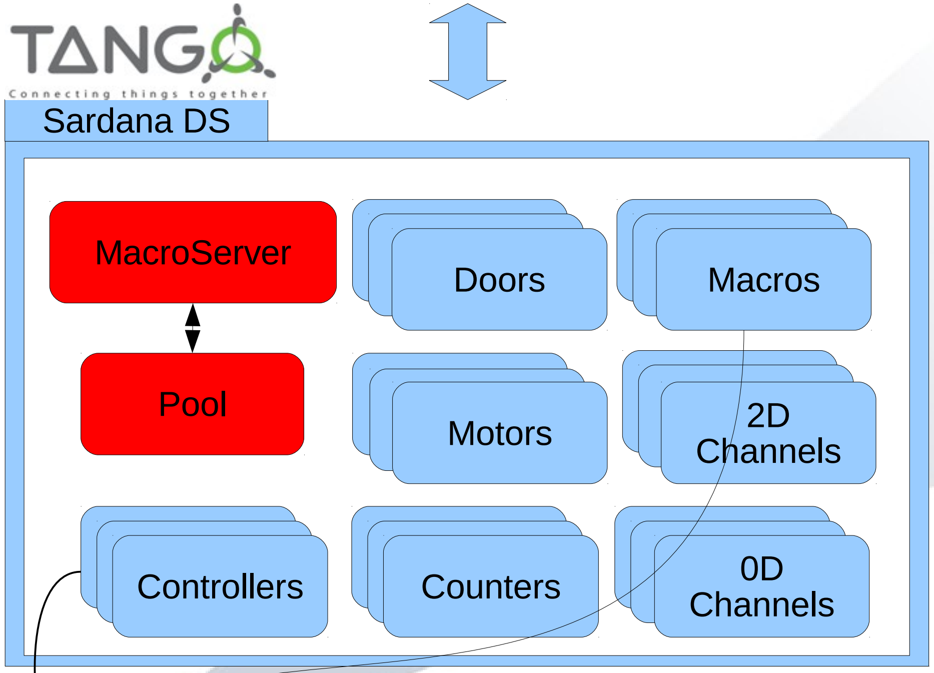
(...or the mid-talk speaker change)

Sardana



```
Python 2.7.1+ (r271:86832, Apr 11 2011, 18:05:24)
Type "copyright", "credits" or "license" for more information.

IPython 0.11 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra
%guiref   -> A brief reference about the graphical user inter
Spock [0]:
Spock [1]: from scipy import special as sp
Spock [2]: x = linspace(0, 20, 100)
Spock [3]: for n in range(4):
   ...:        plot(x, sp.jn(n, x), label=r'$J_%n(x)$' % n)
   ...:
Spock [4]: m1 = Motor("M1")
Spock [5]: m1.position = 1000
Spock [6]:
```



Recent:

- Testing Sardana & Taurus (SEP5)

Future:

- Continuous scan (SEP6)
- HKL (SEP4)
- Lima (SEP2)

Repository controllers: *git://git.code.sf.net/p/sardana/controllers.git*

Repository macros: *git://git.code.sf.net/p/sardana/macros.git*

Next in Sardana: Cont Scan

2014 – … : **continuous scan** development

- requirements analysis & specification writing
- test development to avoid regressions
- incremental & iterative approach

2013 – 2014: continuous scan **proof of concept**

- gathering user requirements

1st – software synchronization

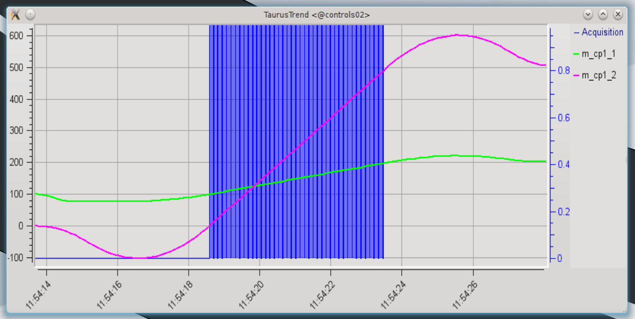
2nd – hardware synchronization (external data buffering)

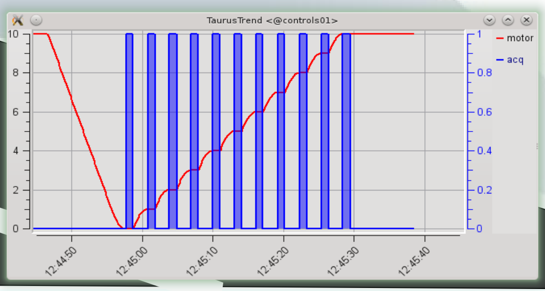
3rd – hardware synchronization only (online data)

(used at ALBA; available in *develop* and *sep6* branches)

< 2013: generic **step scan** framework

- sequential motion & acquisition control
- straightforward macros; multiple recorders
- commissioned and in production





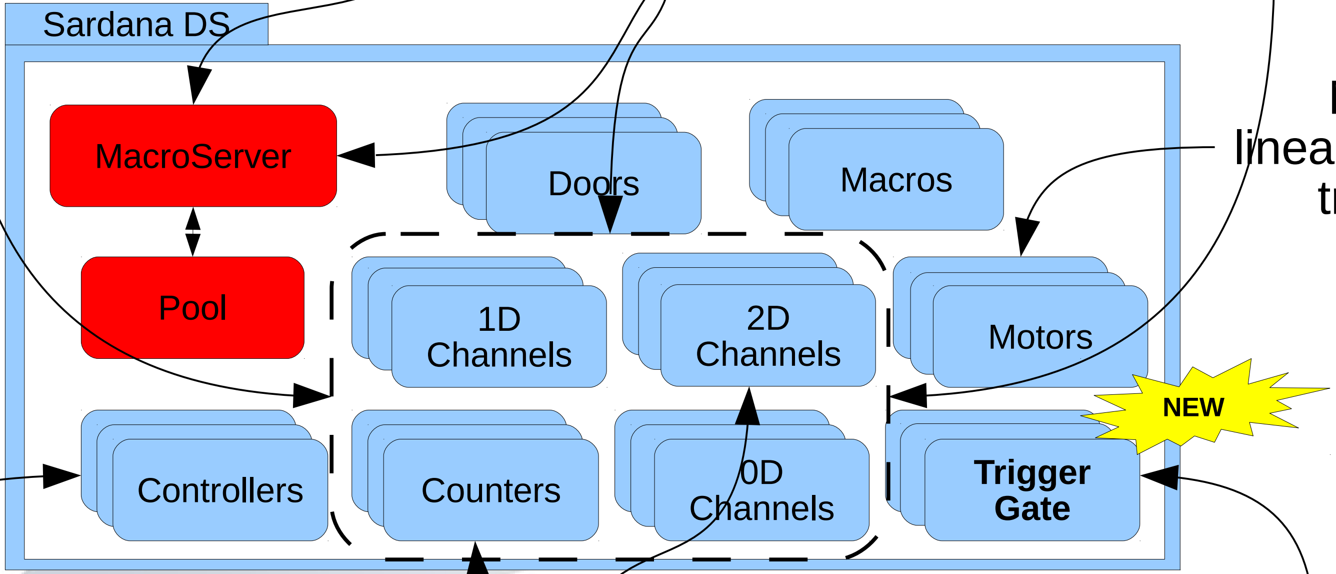
Next in Sardana: Cont Scan

Measurement Group – relation channel – trigger;
co-existence of software and hardware synchronization

Data collection, merging and storage –
timestamp, raw + interpolated data

Different distribution schemes
require different data transfer

Motors –
linear & non-linear
trajectories



Controller - handle multiple
hardware functionalities

TriggerGate – new element type;
time & position domain;
equidistant and arbitrary

Experimental Channels - aware of multiple
acquisition & report multiple data

Documentation: *http://sourceforge.net/p/sardana/wiki/SEP6*

Proof of Concept repository: *develop + sep6 branches*

SEP6
Sardana Enhancement Proposal

Next in Sardana: HKL

Thanks to Teresa Nuñez and Frédéric Picca!

Introduction:

- Allows control over different types of diffractometer
  e.g. Diffrac2C, Diffrac4C, Diffrac6C, DiffracK6C
- Uses HKL library developed by - Frédéric Picca, SOLEIL
  (repo: git://repo.or.cz/hkl.git, available for debian)

Status:

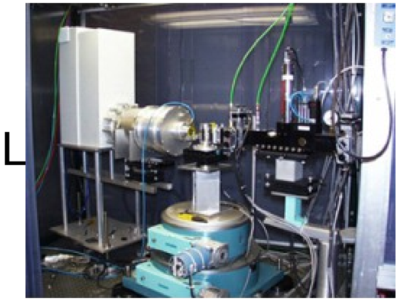
- First implementation ready to be tested

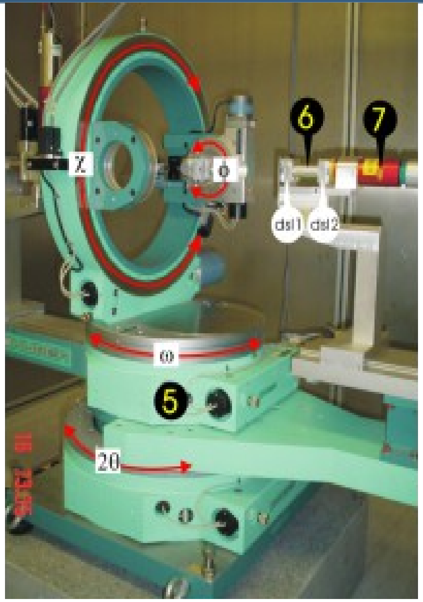
Description:

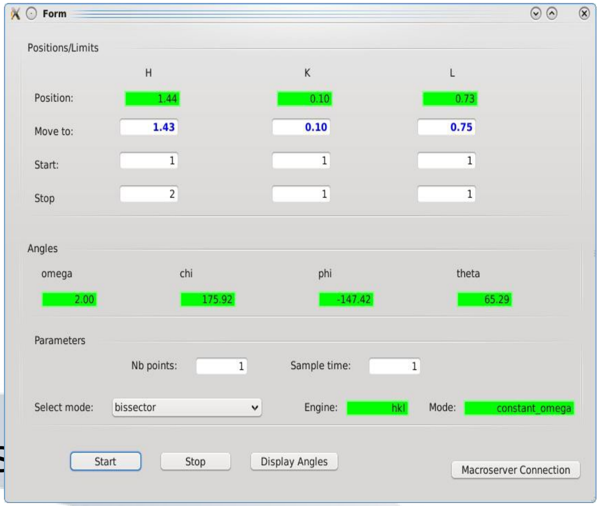
- PseudoMotor controllers instead of integration in the Sardana core
- Base class with all calculation
- Derived classes with the axes definition
- Taurus GUIs connected to the controller, pseumotors, door:
  hkl scan, diffractometer alignment

Plans:

- Test the current implemetnation, discuss about alternative solutions
- Integrate and distribute with Sardana






Form

Positions/Limits

	H	K	L
Position:	1.44	0.10	0.73
Move to:	1.43	0.10	0.75
Start:	1	1	1
Stop	2	1	1

Angles

omega	chi	phi	theta
2.00	175.92	-147.42	65.29

Parameters

Nb points: 1 Sample time: 1

Select mode: bissector Engine: hkl Mode: constant_omega

Documentation: *http://sourceforge.net/p/sardana/wiki/SEP4/*

Repository: *git://git.code.sf.net/u/tere29/sardana*

SEP4
Sardana Enhancement Proposal

Next in Sardana: Lima

Thanks to Gabriel Jover and LIMA team!

Introduction:

- Allows control over 2D detectors in a standard way,
- Uses Lima library developed by ESRF

(repo: https://github.com/esrf-bliss/Lima.git; doc: http://lima.blissgarden.org)

Status:

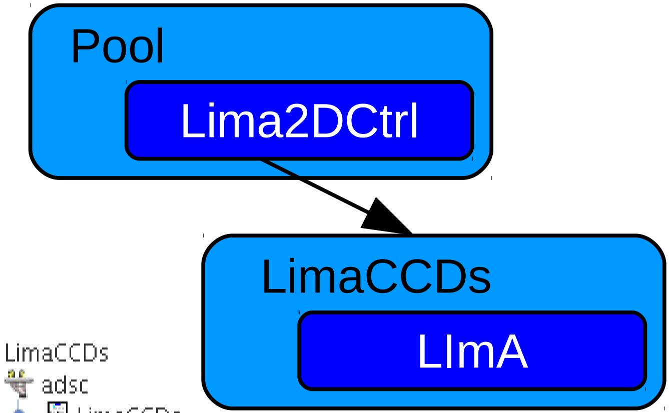
- First implementation in production at ALBA

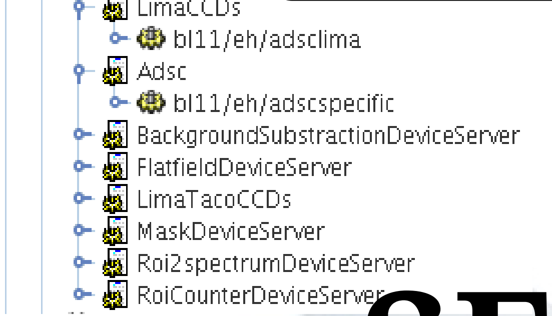
Description:

- Current implementation uses Lima via LimaCCDs Tango DS
- Allow only a single axis
- ReadOne return the image (obtained with getImage cmd)

Plans:

- Use DataSource to pass image location instead of data
- Use Lima library directly instead of DS
- Integrate and distribute with Sardana





```
LimaCCDs
  adsc
    LimaCCDs
      bl11/eh/adsclima
    Adsc
      bl11/eh/adscspecific
    BackgroundSubstractionDeviceServer
    FlatfieldDeviceServer
    LimaTacoCCDs
    MaskDeviceServer
    Roi2spectrumDeviceServer
    RoiCounterDeviceServer
```

Documentation: *http://sourceforge.net/p/sardana/wiki/SEP2*
Repository: *git://git.code.sf.net/p/sardana/controllers.git*

SEP2
Sardana Enhancement Proposal

Testing Framework

Thanks to Marc Rosanes and ALBA team!

Motivation:

- Allow **CI** and **TDD**
- Avoid **regression**
- Help **integration** managers to evaluate contributions

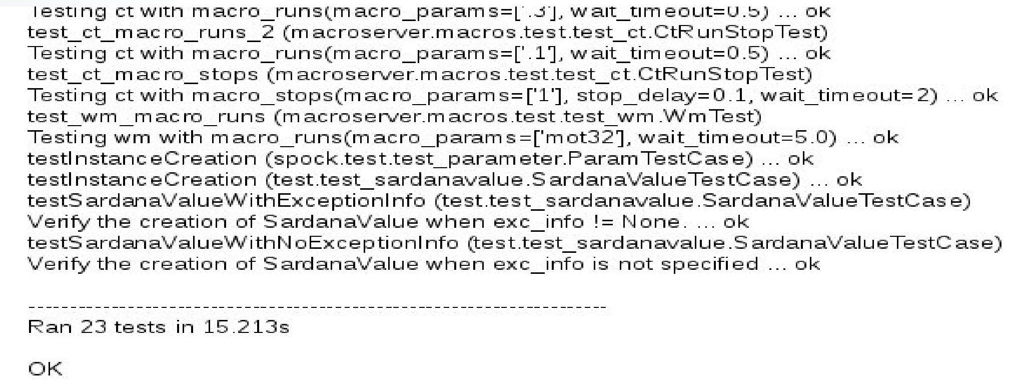
Scope:

- Establish guidelines and examples
- Do not cover with tests the whole system

Description:

- PyUnit testing framework – unittest module
- documentation standard
- test location: tests, utilities, resources
- generic macro testing: Run, Stop, Fail
- generic widget testing
- Test utilities: server control; sardemo
- Autodiscovery test suites

```
@testRun(macro_params=['mot1', '0', '3', '10', '.1'])
@testRun(macro_params=['mot2', '3', '0', '10', '.1'])
@testStop(macro_params=['mot1', '0', '5', '3', '.1'])
class AscanctTest(RunStopMacroTestCase, TestCase):
    macro_name = 'ascanct'
    door_name = 'door1'
```



```
Testing ct with macro_runs(macro_params=['.3'], wait_timeout=0.5) ... ok
test_ct_macro_runs_2 (macroserver.macros.test.test_ct.CtRunStopTest)
Testing ct with macro_runs(macro_params=['.1'], wait_timeout=0.5) ... ok
test_ct_macro_stops (macroserver.macros.test.test_ct.CtRunStopTest)
Testing ct with macro_stops(macro_params=['1'], stop_delay=0.1, wait_timeout=2) ... ok
test_wm_macro_runs (macroserver.macros.test.test_wm.WmTest)
Testing wm with macro_runs(macro_params=['mot32'], wait_timeout=5.0) ... ok
testInstanceCreation (spock.test.test_parameter.ParamTestCase) ... ok
testInstanceCreation (test.test_sardanavalue.SardanaValueTestCase) ... ok
testSardanaValueWithExceptionInfo (test.test_sardanavalue.SardanaValueTestCase)
Verify the creation of SardanaValue when exc_info != None. ... ok
testSardanaValueWithNoExceptionInfo (test.test_sardanavalue.SardanaValueTestCase)
Verify the creation of SardanaValue when exc_info is not specified ... ok

----------------------------------------------------------------------
Ran 23 tests in 15.213s

OK
```

Documentation: *http://sourceforge.net/p/sardana/wiki/SEP5*

Repository: already in develop!

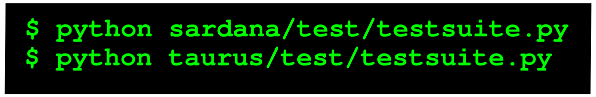
SEP5

Sardana Enhancement Proposal

Jul14 release

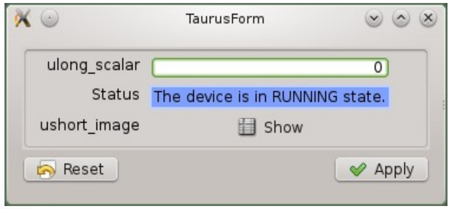
- Bug fixes

- SEP5 (testing)



```
$ python sardana/test/testsuite.py
$ python taurus/test/testsuite.py
```

- SEP9 (compact read write widgets)



TaurusForm

ulong_scalar 0

Status The device is in RUNNING state.

ushort_image Show

- SEP10 (sardana – taurus isolation)

*http://sf.net/p/**sardana***
*http://sf.net/p/**tauruslib***

git **[wiki]** lists

- Improved synoptic

Thanks to Valentin Valls!

- taurus.external

```
from taurus.external.qt import Qt
from taurus.external import unittest
```

Download from: https://pypi.python.org/pypi/sardana
https://pypi.python.org/pypi/taurus

Thank you!

Questions?