



SARDANA – Scientific SCADA Suite

by Guifré CUNÍ, Grzegorz KOWALSKI, Zbigniew RESZELA
on behalf of **Sardana and Taurus Communities**

Sardana Workshop @ ICALEPCS2019, New York, 6th October 2019

Outline

- What is Sardana Suite?
- Sardana Suite overview
- Software architecture & layers
- Sardana Community

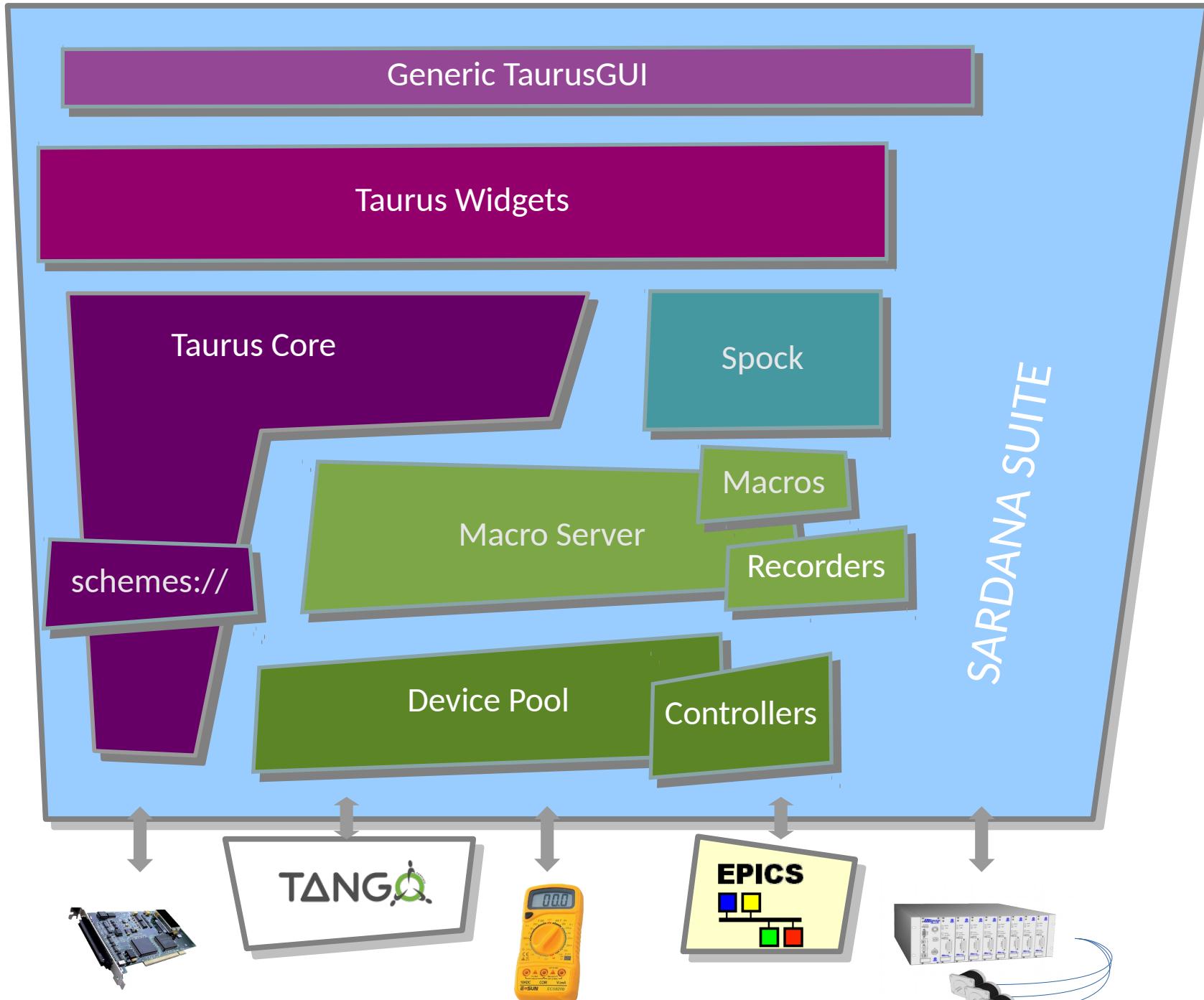
What is Sardana Suite?

What is Sardana Suite?

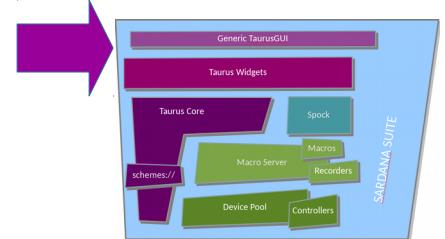
- Scientific laboratory use cases *e.g.*, synchrotron beamline, accelerator, laser labs
 - Scalability
 - Distributed architecture
- User interfaces to the laboratory equipment
 - Common look & feel
 - Data sources access
- Control and supervision of experiment procedures *e.g.*, DAQ, motion, scan
 - Data handling
- HW/instrument management
 - Abstract interfaces
 - Optimization and synchronization
- Extendability & Usability
 - Python plugins
- Flexibility (Configure, don't program!)
- Modularity
- Actively community driven

Sardana Suite overview

Sardana Suite overview



Generic Taurus GUI



User perspectives

Macro execution widgets

1D & 2D plots

Forms with devices and/or attributes

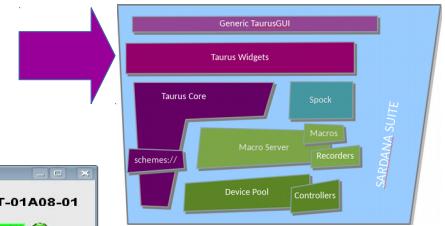
Interactive synoptic

External widgets

BL22 (ALBA) GUI created with the TaurusGUI framework

TaurusGUI framework for creating complete GUIs **without programming a single line of code!**

Taurus Widgets



- Qt based graphical widgets: generic forms, plots, ...
- Sardana specific widgets: macro executor, motor, experiment configuration, scan plots, ...

Code Editor (right):

```

ms/v3/1
  Macro libraries
    training
      where_moveable_2
      where_moveable
      snapshot_ccd
      set_scandid
      scan_info
      set_scandid
      moveable
      move_energy
      move2
      mandelbrot
      hello_world_2
      fixed_ascan
      fft_freq
      cosine
      test01
      submacros
      standard
      sequence
      scans
      scan
      parameters
      aca
      lists
      ioregister
      hooks
      funcs
      expert
      env
      demo
      communication

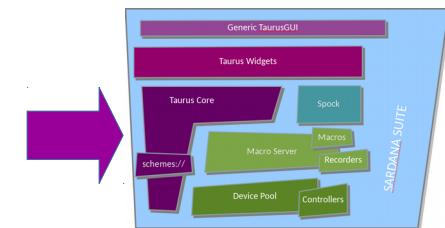
```

```

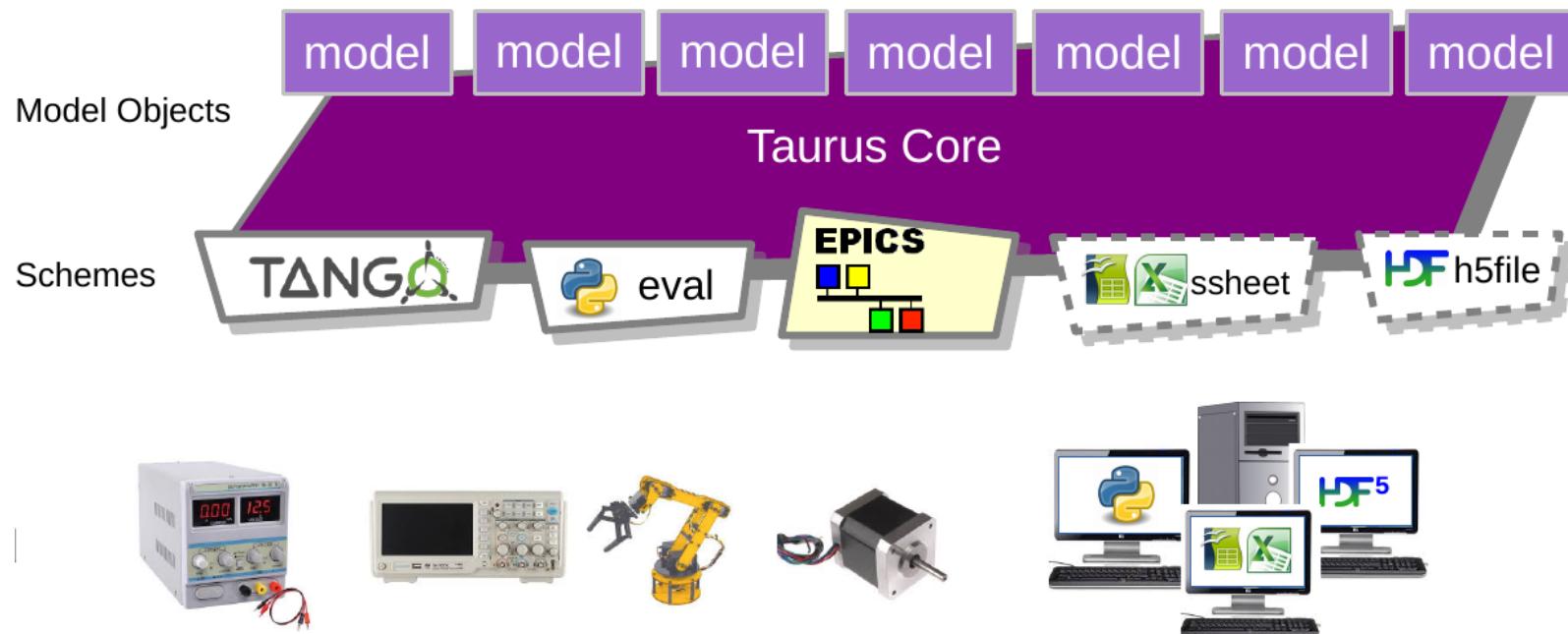
71 """This macro changes the current energy"""
72 E = self.getMoveable("Energy")
73 E.move(energy)
74 self.output("Energy is now at %s", E.getPosition())
75
76
77 @macro([ ["ccd", Type.String, None, "ccd name"],
78         [ "fname", Type.String, None, "filename to save" ] ])
79 def snapshot_ccd(self, 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             1" ],
87         [ "p2", Type.Float, None, "absolute position of mot
88             2" ] ])
89 def move2(self, p1, p2):
90     motion = self.getMotion(["mot01", "mot02"])
91     motion.move([p1, p2])
92
93 @macro([ [ "moveable", Type.Moveable, None, "moveable to get
94             position" ],
95         [ "position", Type.Float, None, "absolute position" ] ])
96 def where_moveable_2(self, moveable):
97     """This macro prints the current moveable position"""
98     self.wm(moveable)
99
100 @macro([ [ "moveable", Type.Moveable, None, "moveable to mov
101             e" ],
102         [ "position", Type.Float, None, "absolute position" ] ])
103 def move_moveable(self, moveable, position):
104     """This macro moves a moveable to the specified position"""
105     self.mv(moveable, position)
106     self.output("%s is now at %s", moveable.getName(), move
107             able.getPosition())

```

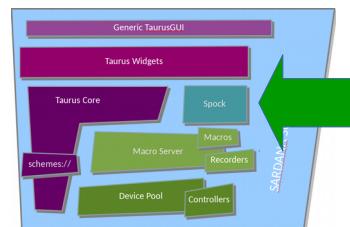
Taurus Core



- “Container of unique models”.
- Schemes provide access to different type of data sources.
- Polling and event mechanism.



Spock



- **Spock** – IPython based Sardana CLI.
- Spock syntax mimics **SPEC** commands.
- Provides total control over the system: executes procedures, interacts with the elements, ...

```
/bin/bash 90x39
tcoutinho@pc151:~/workspace/Spock$ ./spock -p BL98

Spock 7.2.1 -- An interactive Tango client.

Running on top of Python 2.6.6, IPython 0.10 and PyTango 7.2.1dev

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

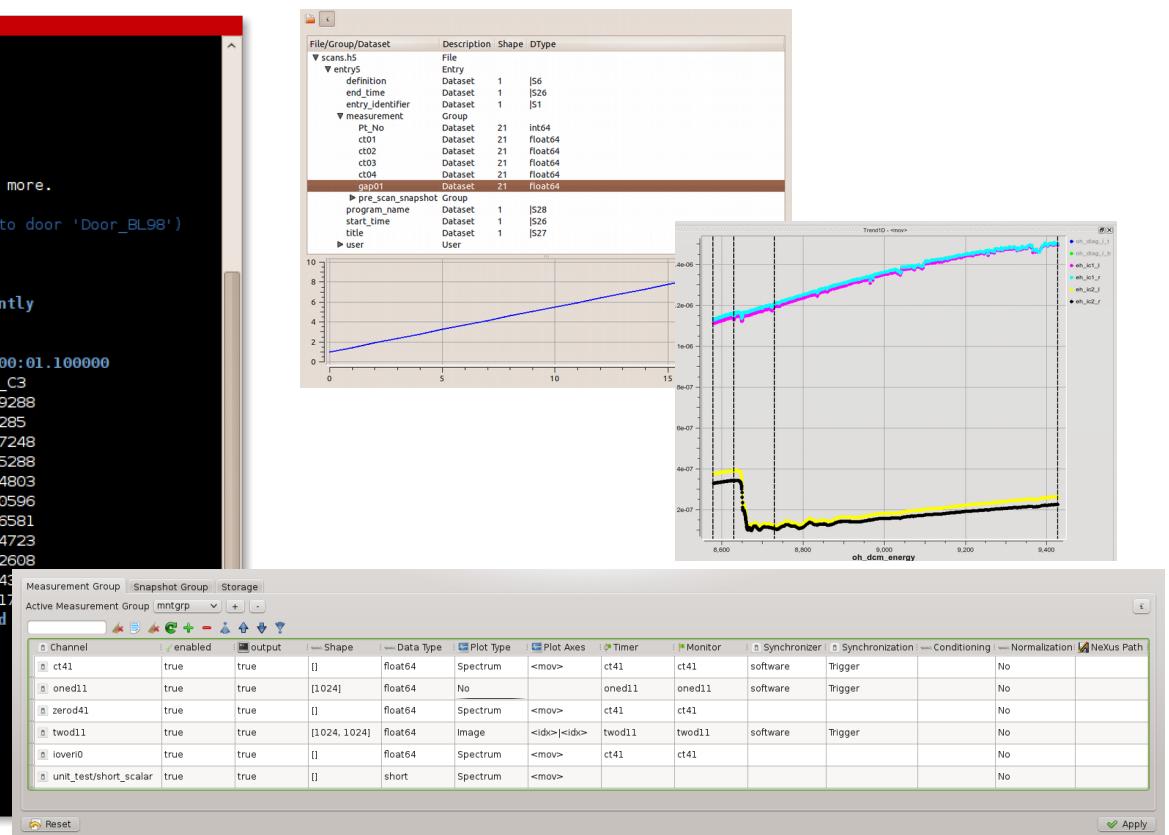
Spock's sardana extension 0.5.0 loaded with profile: BL98 (linked to door 'Door_BL98')

Door_BL98 [1]: %ascan bl98_m1 0 100 10 0.1
ExtraColumns is not defined
ScanDir is not defined. This operation will not be stored persistently
SharedMemory is not defined.
SharedMemory is not defined.
Scan started at Tue Jun 28 18:06:16 2011. It will take at least 0:00:01.100000
 #Pt No    BL98_M1    BL98_Timer    BL98_C1    BL98_C2    BL98_C3
  0        0.0       0.1       0.103096   0.206192   0.309288
  1       10.0       0.1       0.10095    0.2019     0.30285
  2       20.0       0.1       0.102416   0.204832   0.307248
  3       30.0       0.1       0.105096   0.210192   0.315288
  4       40.0       0.1       0.111601   0.223202   0.334803
  5       50.0       0.1       0.113532   0.227064   0.340596
  6       60.0       0.1       0.115527   0.231054   0.346581
  7       70.0       0.1       0.101574   0.203148   0.304723
  8       80.0       0.1       0.117536   0.235072   0.352608
  9       90.0       0.1       0.101459   0.202918   0.3043
 10      100.0      0.1       0.113926   0.227852   0.3417
Scan ended at Tue Jun 28 18:06:33 2011, taking 0:00:16.645132 (dead)

Door_BL98 [2]: wa
Current Positions (user, dial)

BL98_M1    BL98_M2    BL98_MP1
100.0000   43.0000   100.0000
100.0000   43.0000   100.0000

Door_BL98 [3]:
```



MacroServer

Hooks

```
Door_1 [8]: loop 0 10 3
Starting loop
At step 0
running hook with hints=['pre-acq']
In hook 1
At step 3
running hook with hints=['pre-acq']
In hook 1
At step 6
running hook with hints=['pre-acq']
In hook 1
At step 9
running hook with hints=['pre-acq']
In hook 1
Finished loop
```

Input parameters & results & data

SPEC like commands

Plotting

pcapac2014.py

```

13
14 @macro([["moveable", Type.Moveable, None, "moveable to move"], 
15         ["position", Type.Float, None, "absolute position"]])
16 def move(self, moveable, position):
17     """This macro moves a motor to the specified position"""
18     moveable.move(position)
19     self.output("Motor ended at ", moveable.getPosition())
20
21 class loop(Macro, Hookable):
22     """A macro that executes a for loop. It accepts hooks.
23     hints = {'allowsHooks':('pre-move', 'post-move', 'pre-acq', 'post-acq')}
24
25     param_def = [[['start', Type.Integer, None, 'start point'],
26                  ['stop', Type.Integer, None, 'end point']],
27                  ['step', Type.Integer, 1, 'step']]]
28
29     result_def = [['result', Type.Integer, None, 'result']]
30
31     def hook1(self):
32         self.output("In hook 1")
33
34     def run(self, start, stop, step):
35         self.info("Starting loop")
36         self.hooks = [(self.hook1, ["pre-acq"])]
37         for i in xrange(start, stop, step):
38             self.output("At step %d" % i)
39             self.flushOutput()
40
41     def hook_hints_in(self.hooks):
42         for hook, hints in self.hooks:
43             hook()
44         self.info("finished loop")
45         return i
46
47 class hooked_scan(Macro):
48     """An example on how to attach hooks to the various hook points of a scan.
49     """
50     param_def = [
51         ['motor', Type.Moveable, None, 'Motor to move'],
52         ['start_pos', Type.Float, None, 'Scan start position'],
53         ['end_pos', Type.Float, None, 'Scan final position'],
54         ['nr_interv', Type.Integer, None, 'Number of scan intervals'],
55         ['integ_time', Type.Float, None, 'Integration time']]
56
57     def hook1(self):
58         self.info("\thook1 execution")
59
60     def run(self, mot, start, end, nr, intt):
61         self.ascan, pars = self.createMacro("ascan", mot, start, end, nr, intt)
62         self.ascan.hooks = [(self.hook1, ["pre-acq"])]
63         self.runMacro(ascan)
64
65     @property
66     def data(self):
67         return self.ascan.data
68
69     @macro()
70     def ask_number_of_points(self):
71         """Asks user for the number of points"""
72         nb_points = self.input("How many points?", data_type=Type.Integer)
73
74     @macro()
75     def J0_plot(self):
76         """Sample J0 at linspace(0, 20, 200)"""
77         x = linspace(0, 20, 200)
78         y = J0(x)
79         x1 = x[::10]
80
81         y1 = map(j0, x1)
82         self.pyplot.plot(x, y, label=r'$J_0(x)$')
83         self.pyplot.plot(x1, y1, 'ro', label=r'$J_0(x)\backslash\integ(x)$')
84         self.pyplot.title(r'Verify $J_0(x)=\frac{1}{\pi}\int_0^\pi\cos(x\sin\phi)d\phi$')
85         self.pyplot.xlabel('x$')
86         self.pyplot.legend()
87

```

Adding, editing macros at runtime

Interactive macros

System

Data Handler

Macro Server

ShmRec

StdoutRec

JsonRec

NxsScanRec

NxasRec

FioRec

SpecRec

System

ESRF .spec

DESY .fio

Nexus NXas

Nexus NXscan

JSON

> stdout

Macro Server

Generic TaurusGUI

Taurus Widgets

Taurus Core

Spock

Macros

Recorders

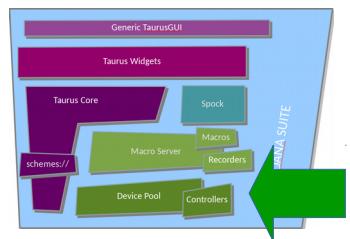
schemes//

Device Pool

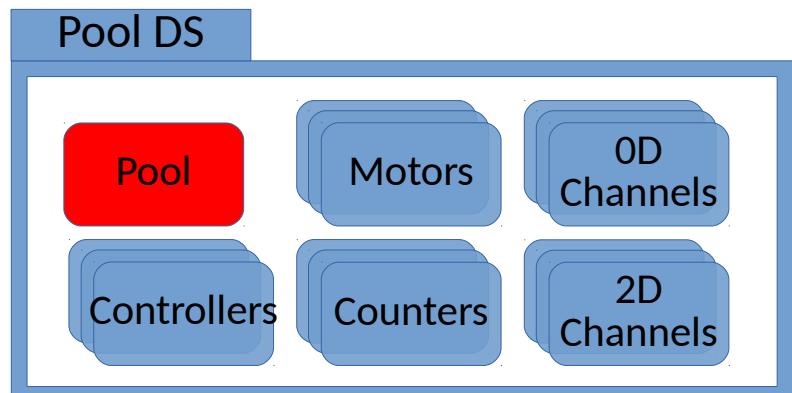
Controllers

Device Pool

12



- All the equipments are interfaced via Device Pool and its **plug-in controller classes (Python)**
- Generic elements' interfaces allow building high level layers on top of them e.g. MeasurementGroup, pseudo elements, generic widgets, etc.

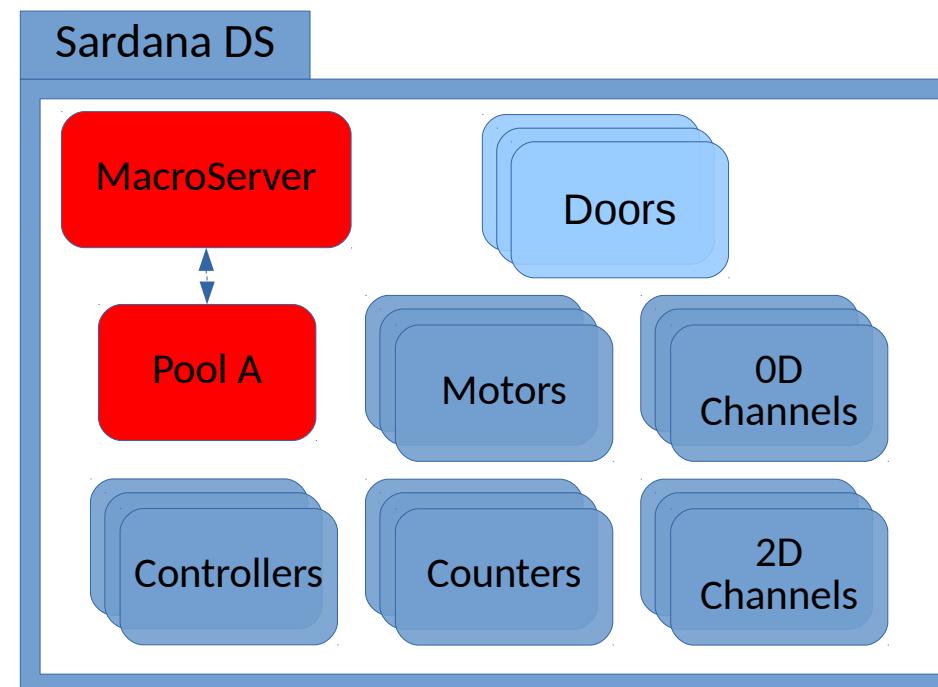
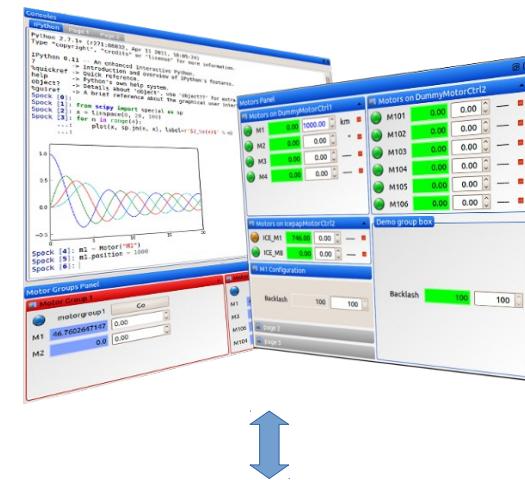


Pool Device Server and its elements

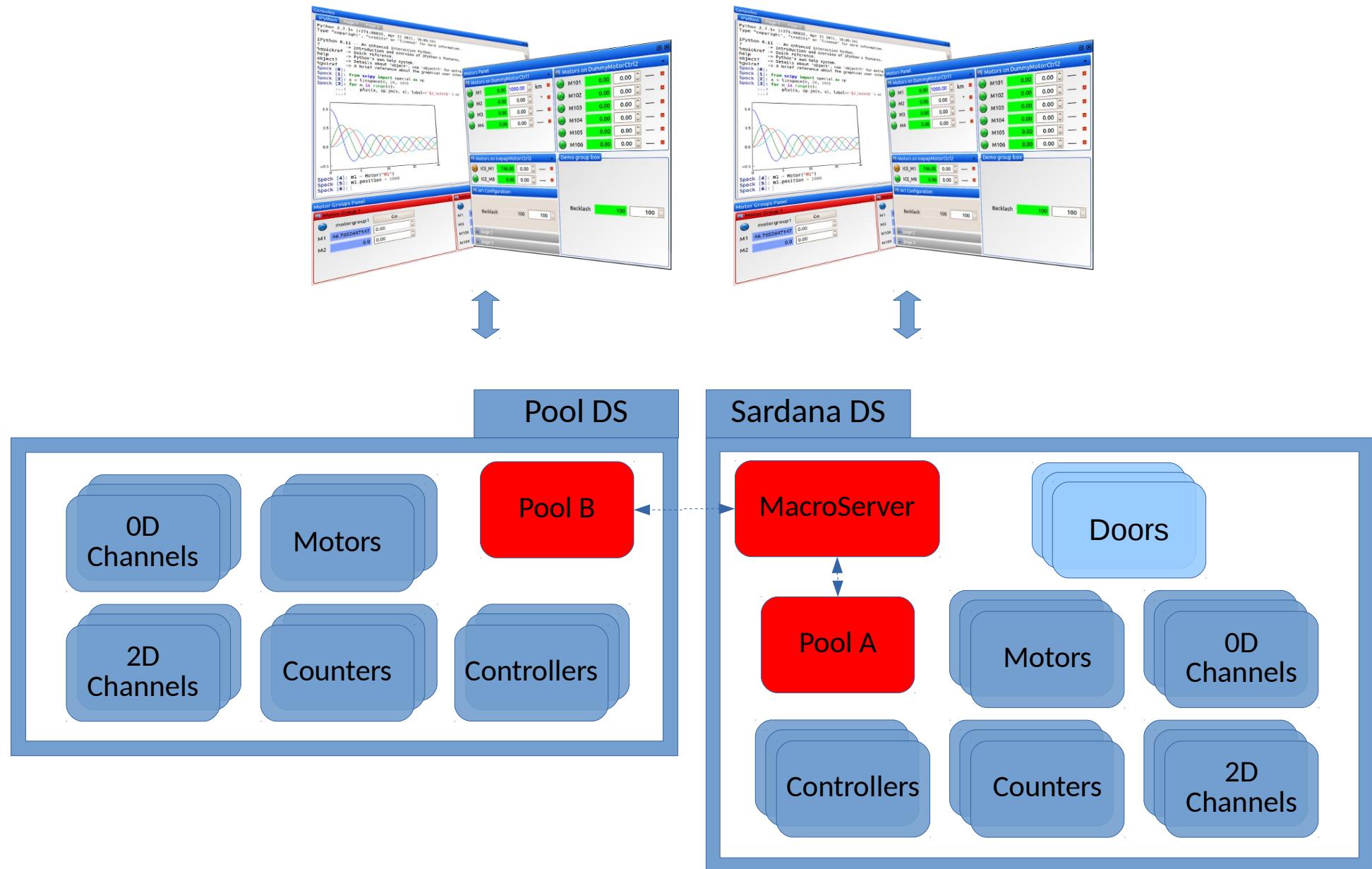
Element Type	Example of application
Motor	stepper, servo or piezo actuator
PseudoMotor	energy, HKL of a diffractometer, slit's gap or offset
CounterTimer	event counter, position measurement, but also... analog to digital converter (ADC), low current electrometer
PseudoCounter	vertical beam position in the X-ray beam position monitor (XBPM)
0DExpChannel	no controlled acq. (software sampling)
1DExpChannel	position sensitive detector (PSD), multichannel analyzer (MCA)
2DExpChannel	CCD camera, 2D X-ray detector
TriggerGate	generators of synchronization signals/events
IORegister	input/output registers of PLC

Software architecture & layers

Client – Server model

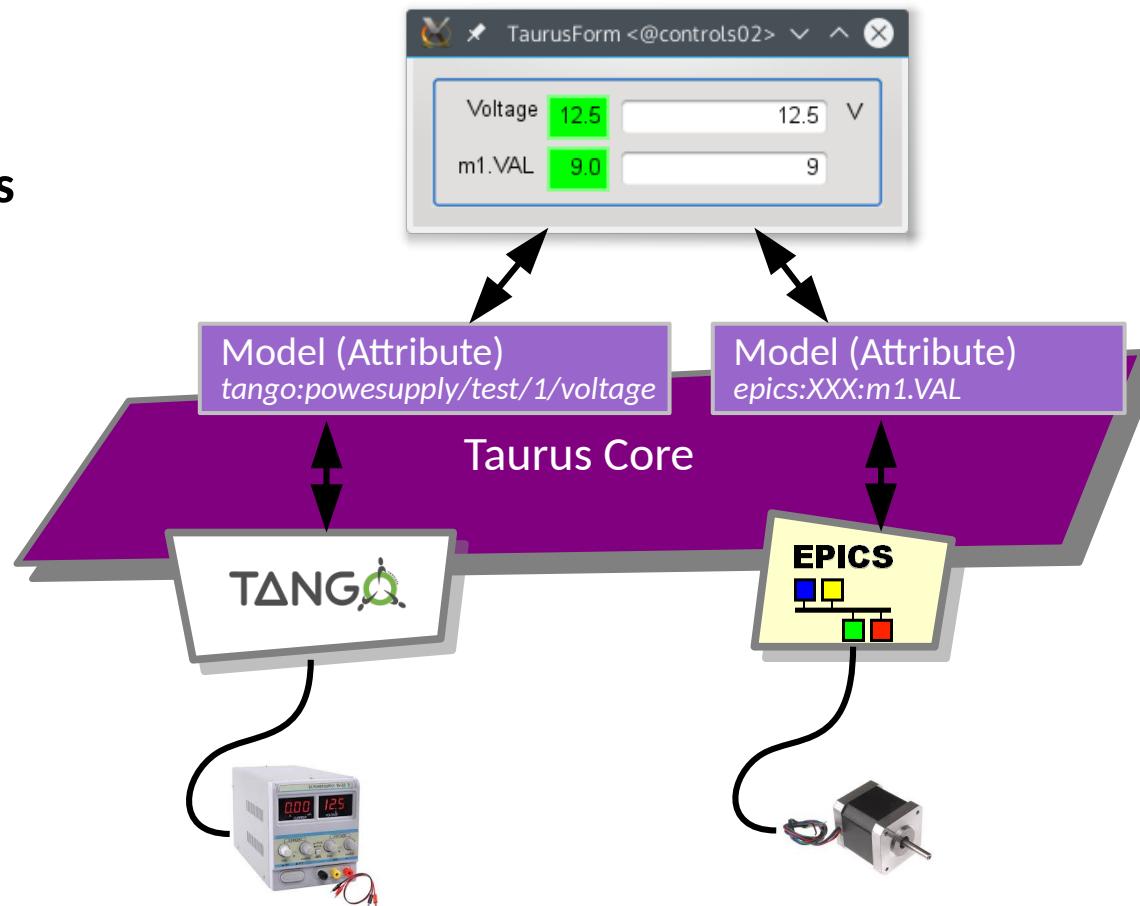


Client – Server model

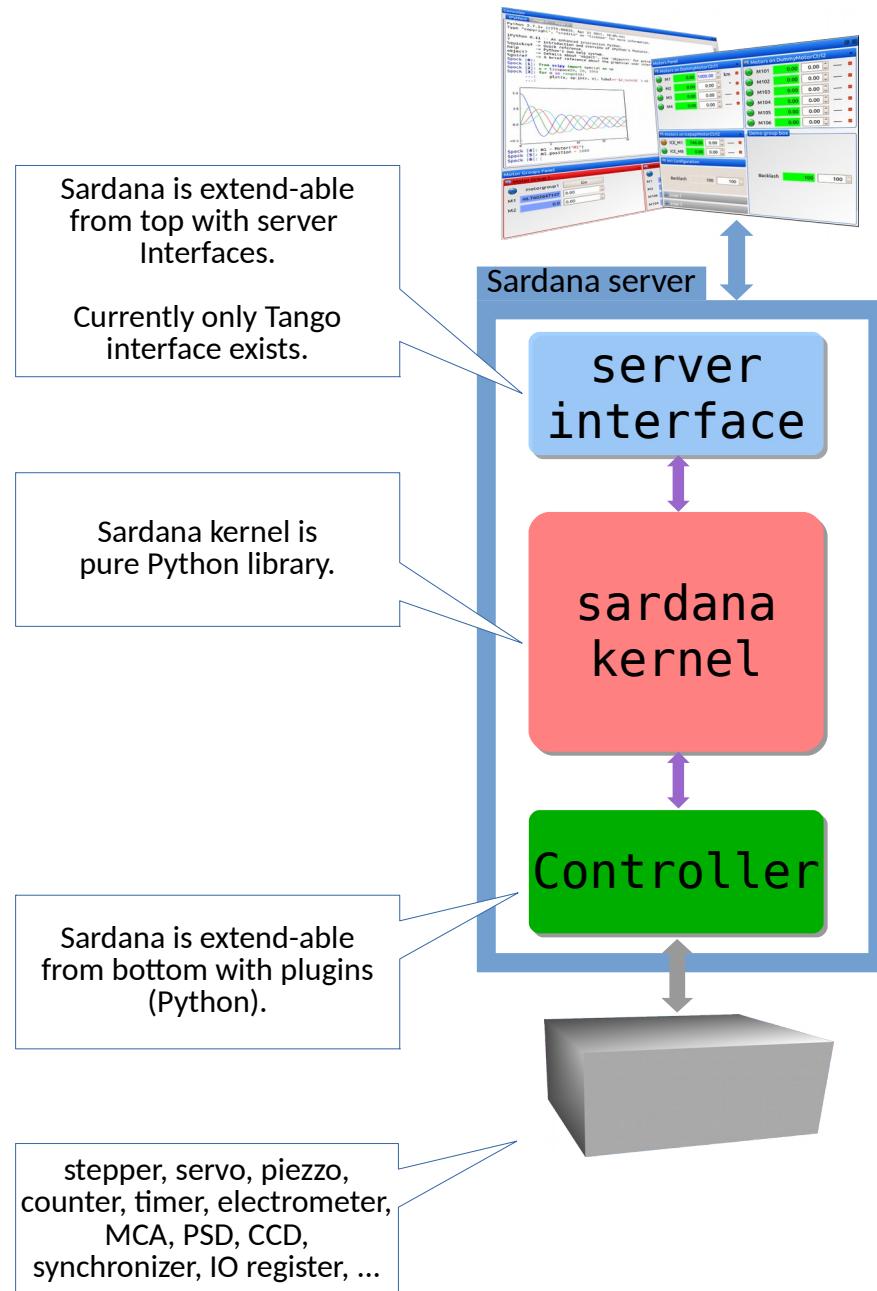


Software layers - Taurus models & schemes

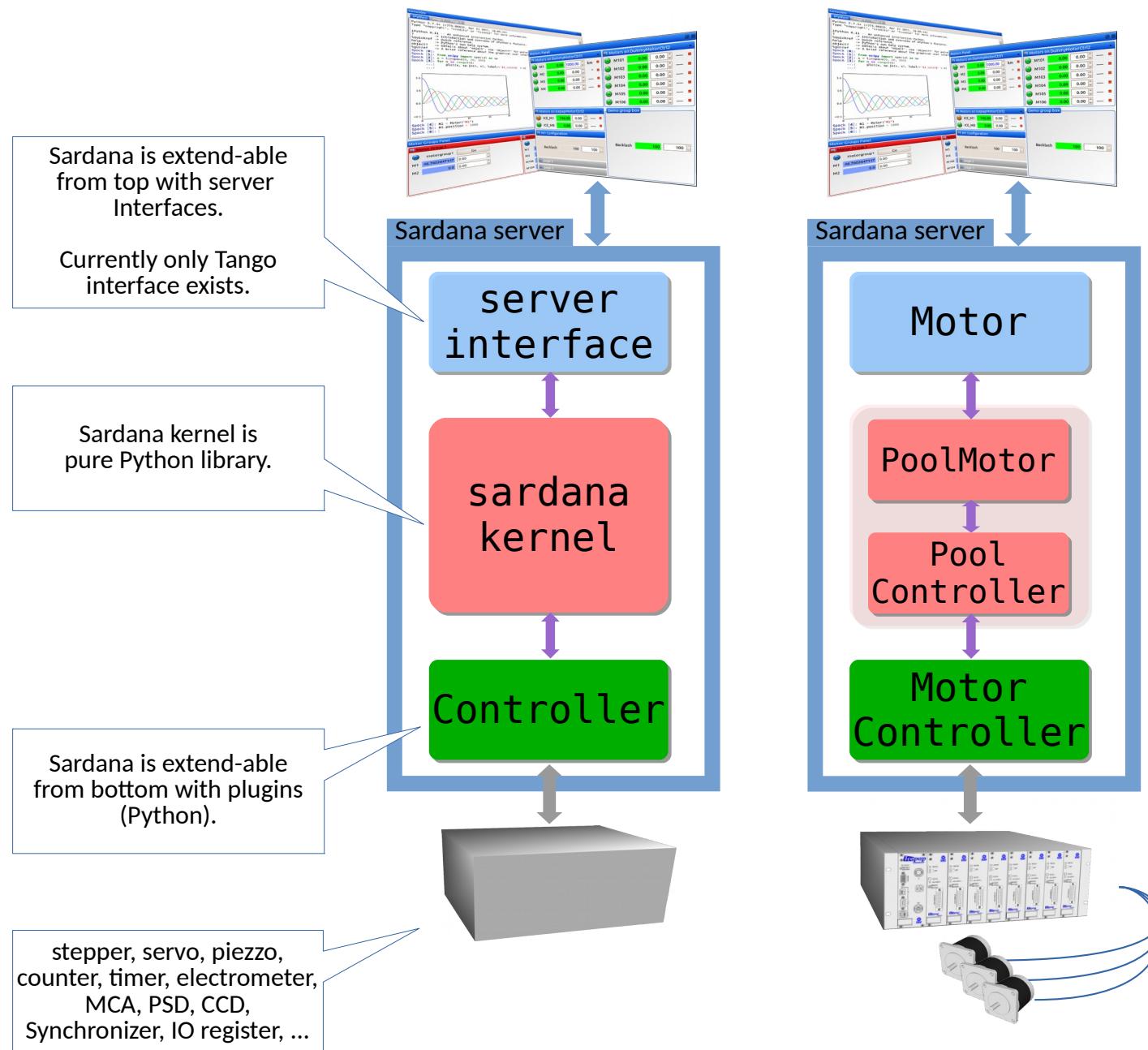
- The model objects are **singletons**
- Model names are **URIs**
- Each scheme provides:
 - A **model factory** for:
 - Authority
 - Device
 - Attribute
 - **Model name validators**



Software layers - Controllers



Software layers - Controllers



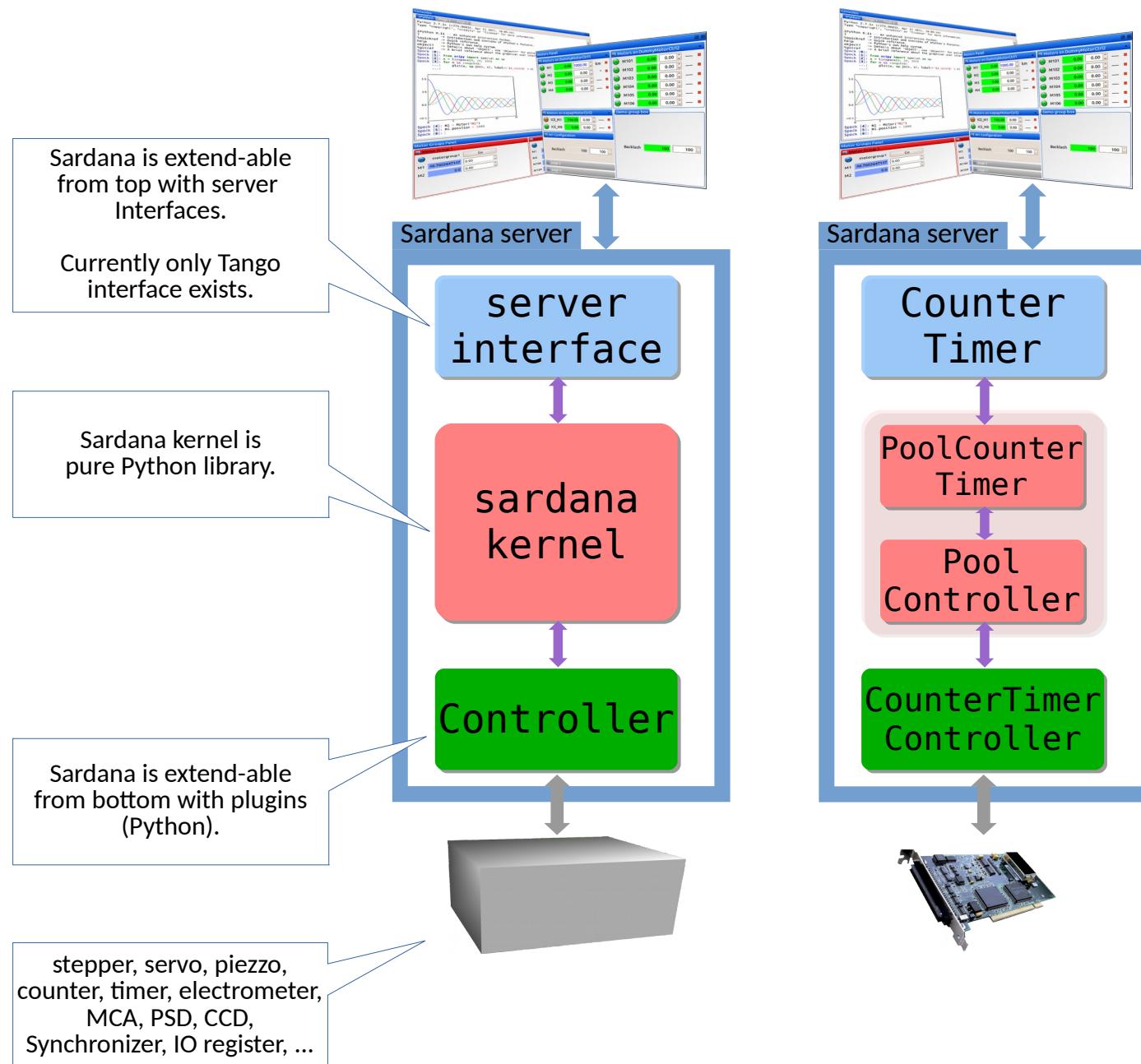
Motor

```
position: float
state: enum
offset: float
sign: int
steps_per_unit
...
```

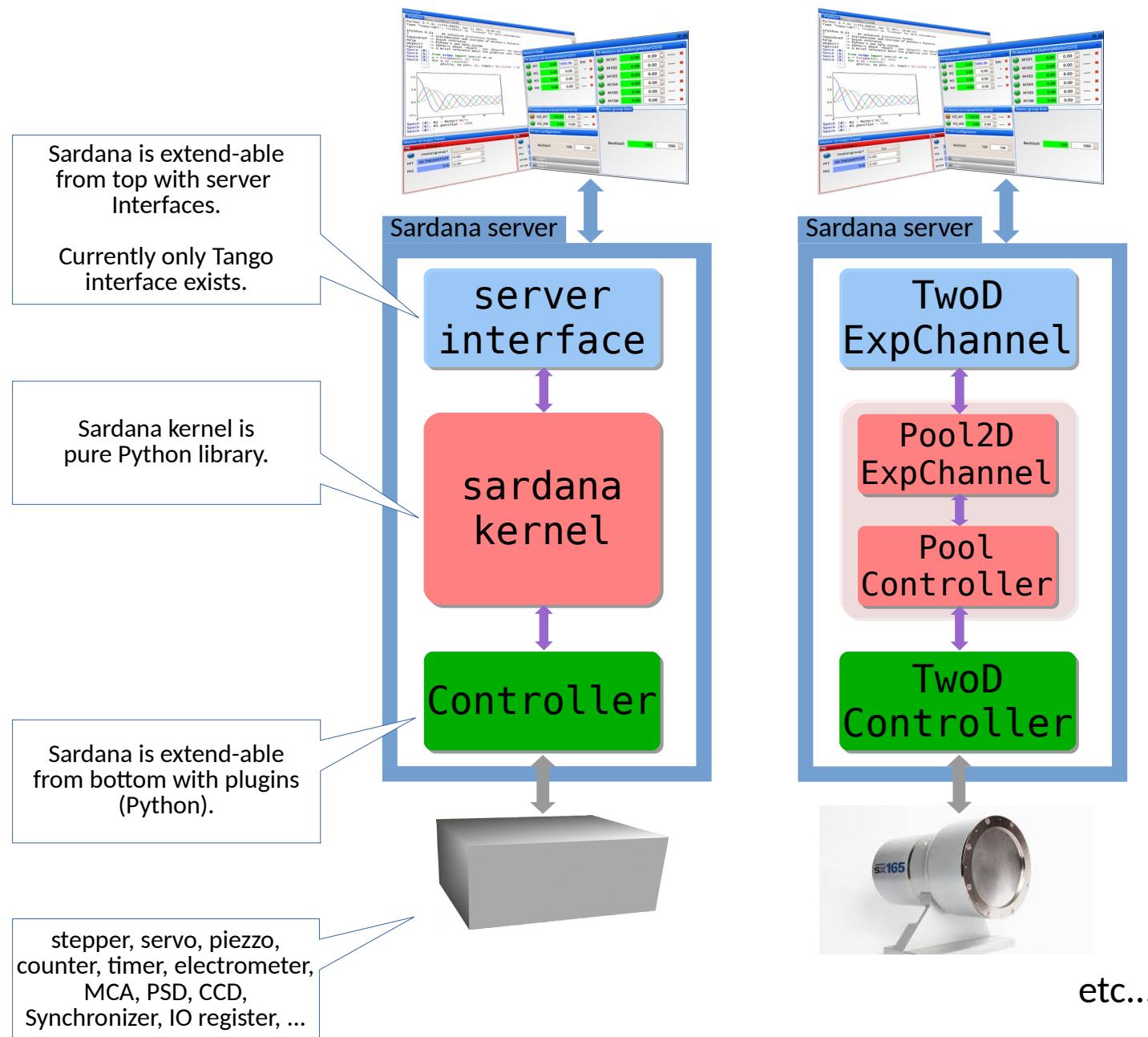
```
Stop()
Abort()
...
```

```
class MyMotorCtrl(MotorController):
    def StateOne(self, axis):
        [...]
    def ReadOne(self, axis):
        [...]
    def StartOne(self, axis, pos):
        [...]
    def AbortOne(self, axis):
        [...]
```

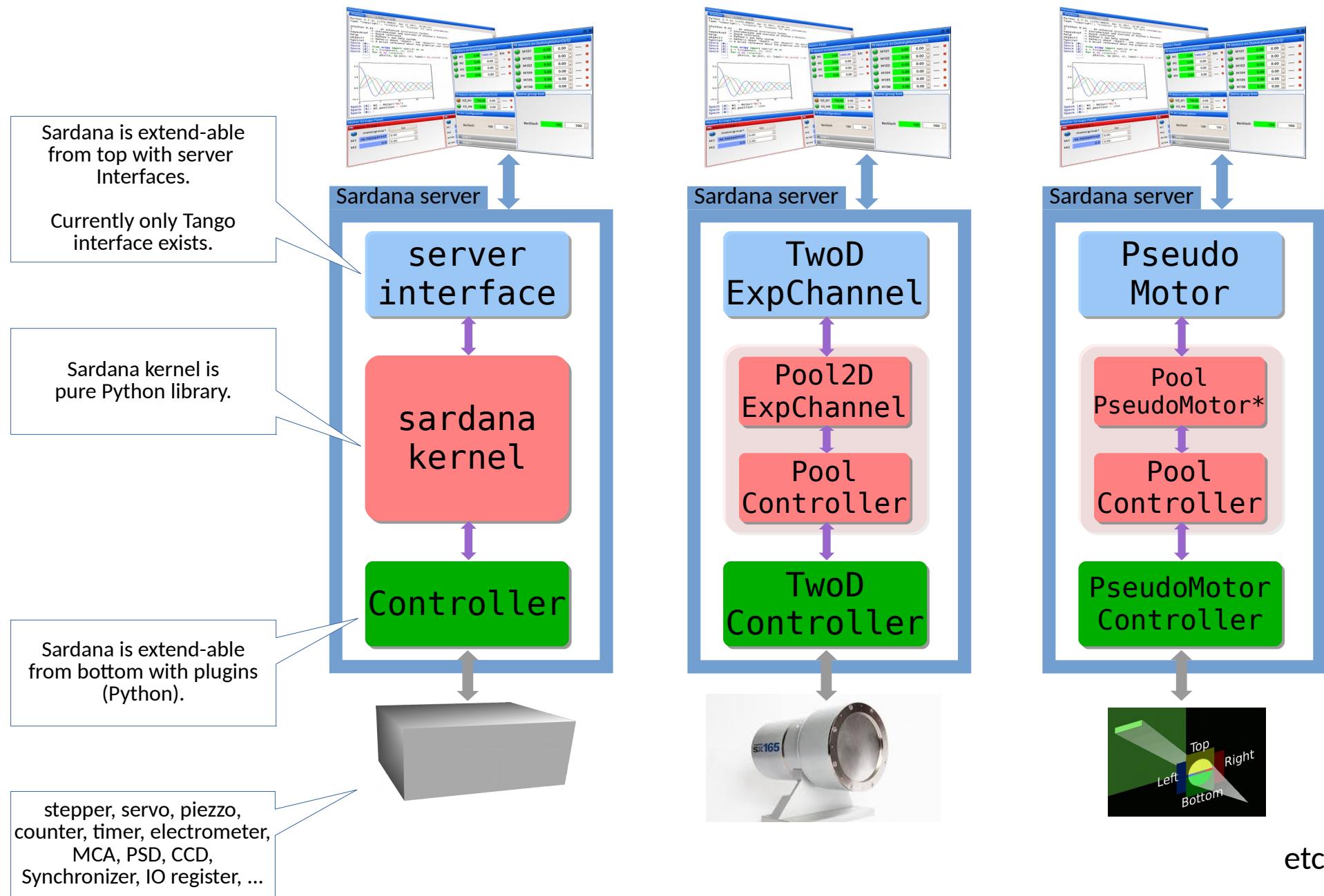
Software layers



Software layers - Controllers

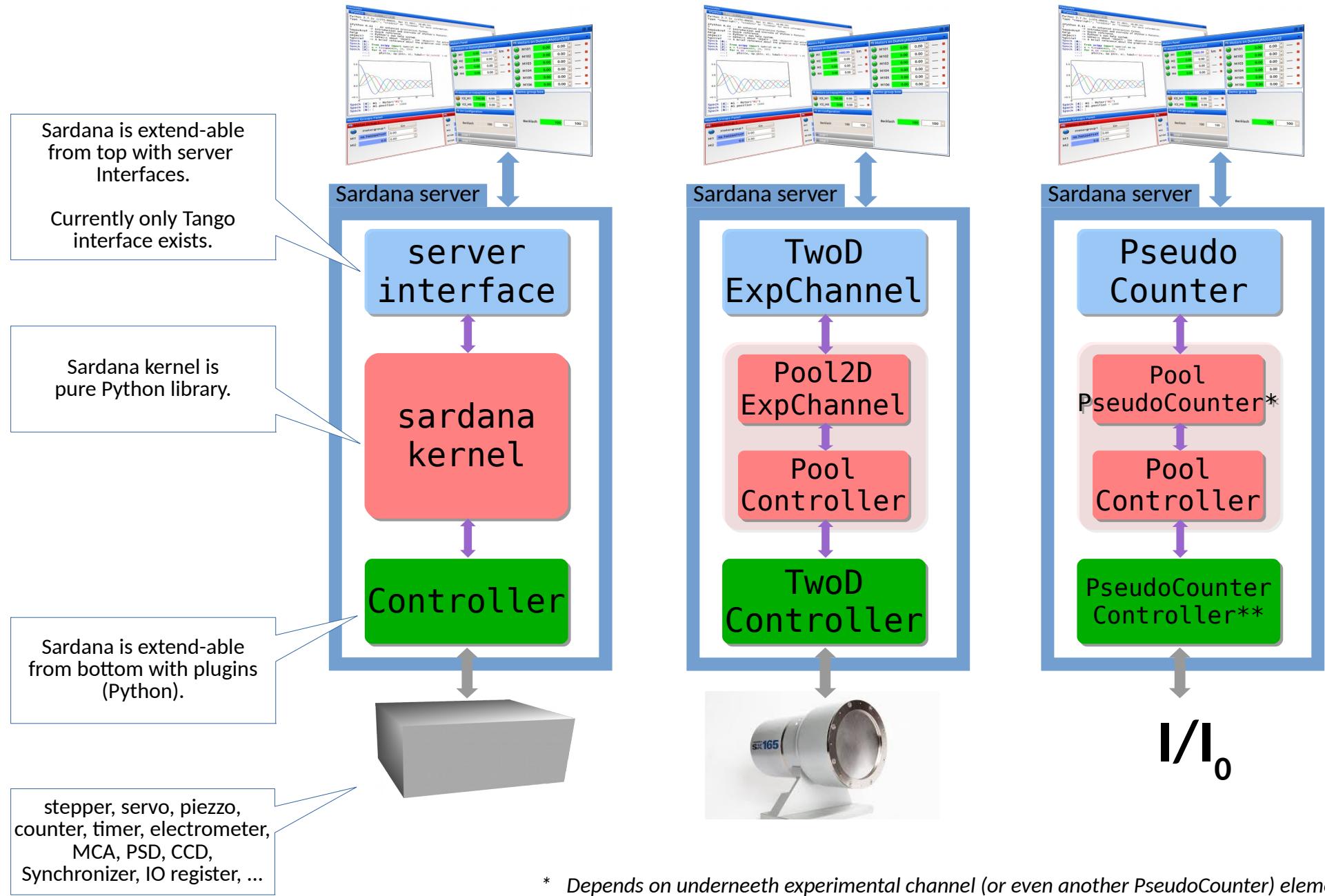


Software layers - Controllers

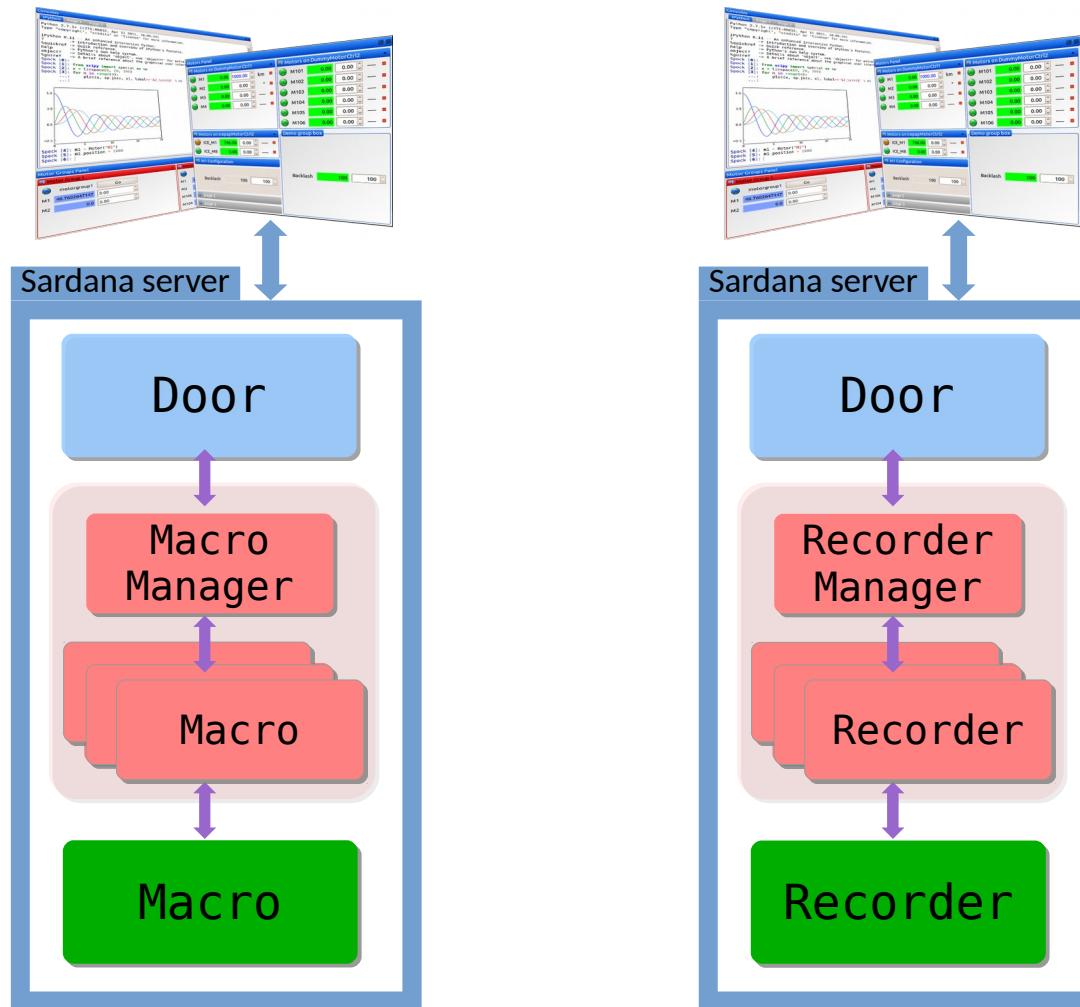


* Depends on underneath PoolMotor (or even another PseudoMotor) elements

Software layers - Controllers

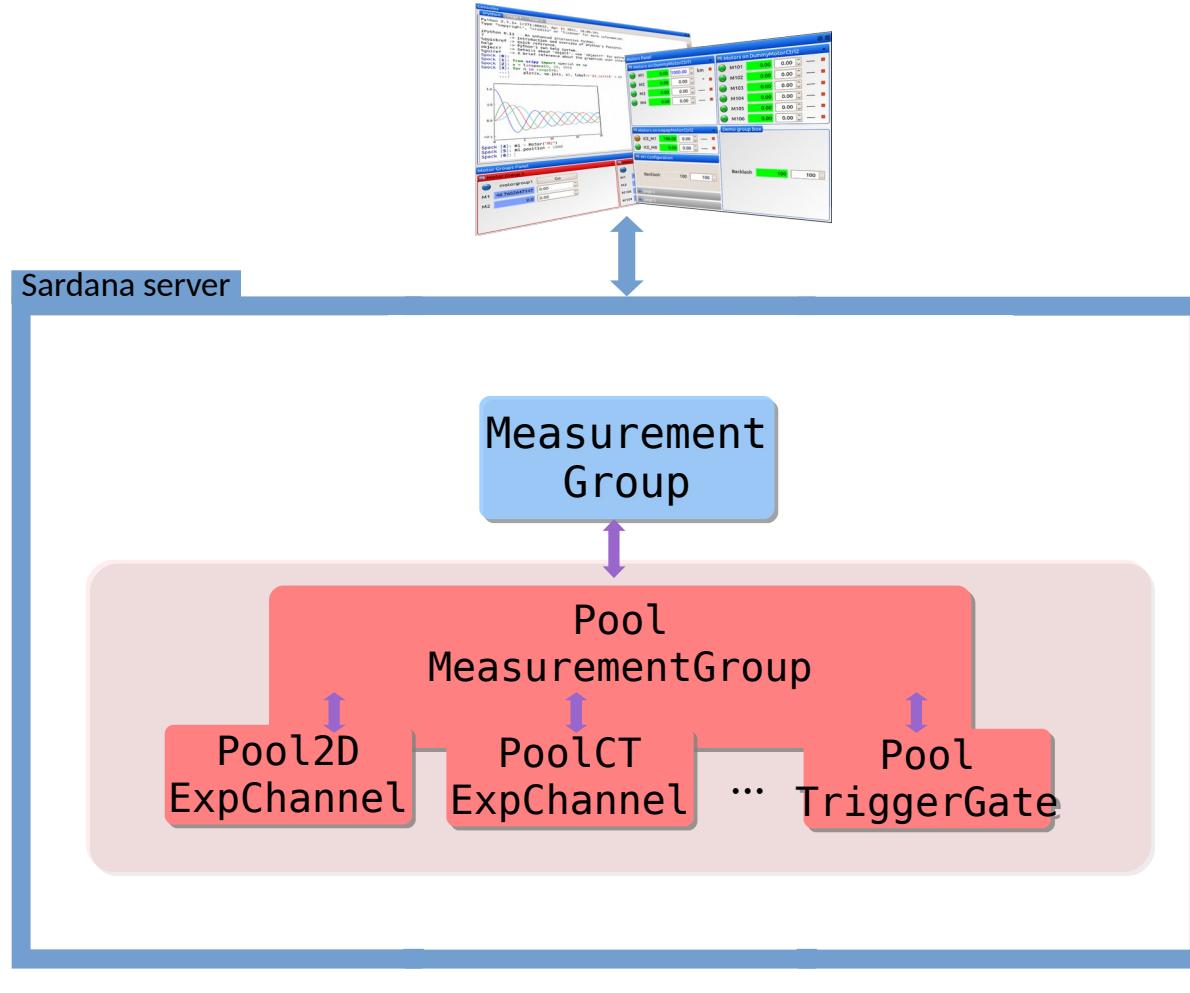


Software layers – Macros & Recorders



- Macros and Recorders are loaded in the Sardana server but **are not long-lived elements**.
- Managers are factories of Recorders or Macros and **create their instances on demand at runtime**.
- Macros may use Recorders to propagate the data to the destination.
- Door is an interface to execute Macros.

Software layers - groups



- Hide the complexity of multi element actions e.g. acquisition or motion, behind a **common interface**.
- Ensure **synchronization** of multi element actions e.g. start of movement and **optimization** of multi element queries e.g. read motor positions.

Other software related aspects...



Travis CI



AppVeyor

GitHub Pages



debian



All software layers have **automatic tests** (unit, integration or acceptance tests) run in CI (Travis CI & AppVeyor). We practice as much as possible TDD.

Static code analysis (flake8) and **documentation builds** are also part of the CI.

Sardana and Taurus form part of the **Debian releases**.

Sardana and Taurus **Docker images** are provided for newcomers.

Sardana and Taurus Communities

Sardana applications



DESY:

- 14 beamlines (all in production)

MAXIV:

- 3 beamlines + 8 beamlines in construction
- used in the Accelerator for motion control

SOLARIS:

- 2 beamlines (both in production)
- planned to be used in the Accelerator

Max Born Institute:

- 2 laboratory setups for time-resolved experiments studying magnetism using femtosecond light pulses (MOKE and XMCD)
- Sardana within Tango and EPICS environment!

ESRF:

- used in the Accelerator - MacroServer only (control the automatic topup system; start/stop the equipments in order to send new electrons in the storage ring)

LPS Orsay (Paris)

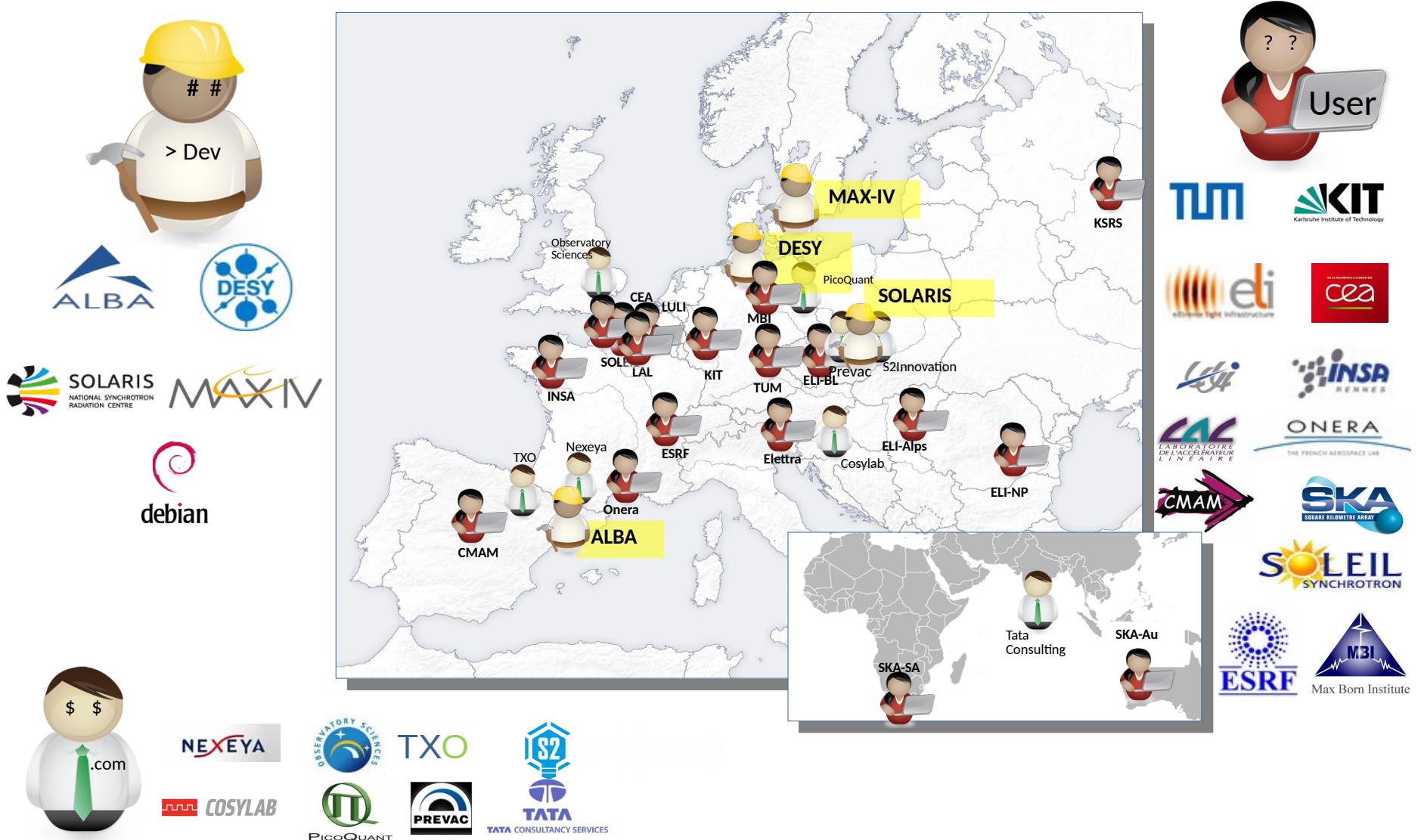
- control an Eulerian-4 circle diffractometer to perform scans

ALBA:

- 8 beamlines (in production) + 4 more beamlines (planned/in construction)
- Optics and Material Science laboratories
- used in the Accelerator for: motion control, scans and operation



Taurus Community



Community – Tools & Events



Docs: <http://www.sardana-controls.org>
<http://www.taurus-scada.org>

GitHub

Projects: <https://github.com/sardana-org/sardana>
<https://github.com/taurus-org/taurus>

SEP TEP

Sardana Enhancement Taurus Enhancement
Proposal Proposal

SEP index: <https://sardana-controls.org/sep/index.html>
TEP index: <https://taurus-scada.org/tep/index.html>



Videoconf: <https://meet.jit.si/sardana>
<https://github.com/sardana-org/sardana-followup>



Plugins Catalogue: <https://github.com/sardana-org/sardana-plugins>

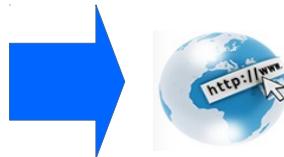


Training and workshops: <https://github.com/sardana-org/sardana-training>
(satellite to Tango meetings or ICALEPCS conference)



Announcements: sardana-users@lists.sourceforge.net
tauruslib-users@lists.sourceforge.net
Coordination: sardana-devel@lists.sourceforge.net
tauruslib-devel@lists.sourceforge.net

Community – Tools & Events

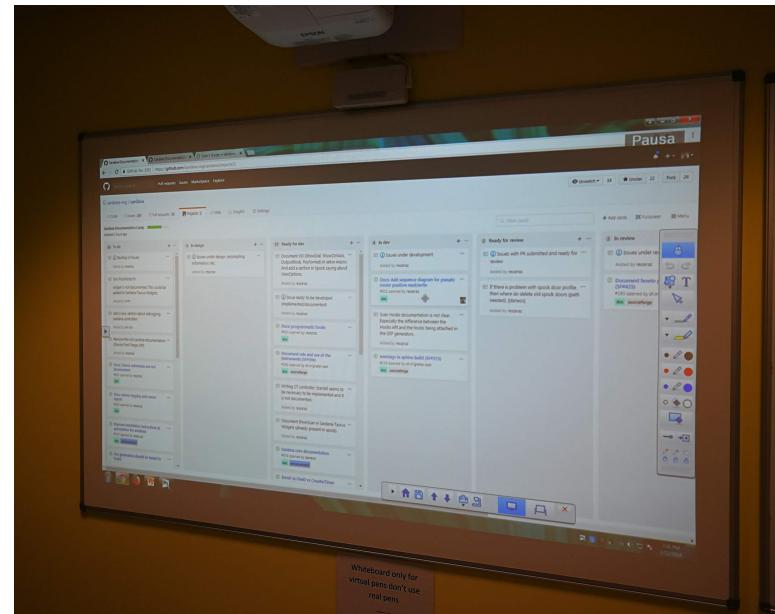


Docs: <http://www.sardana-controls.org>
<http://www.taurus-scada.org>

GitHub

SEP TEP

Sardana Enhancement Taurus Enhancement
Proposal Proposal

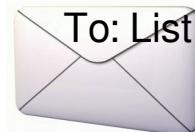


Community – Tools & Events



Projects: <https://github.com/sardana-org/sardana>
<https://github.com/taurus-org/taurus>

SEP TEP
 Sardana Enhancement Taurus Enhancement
 Proposal Proposal



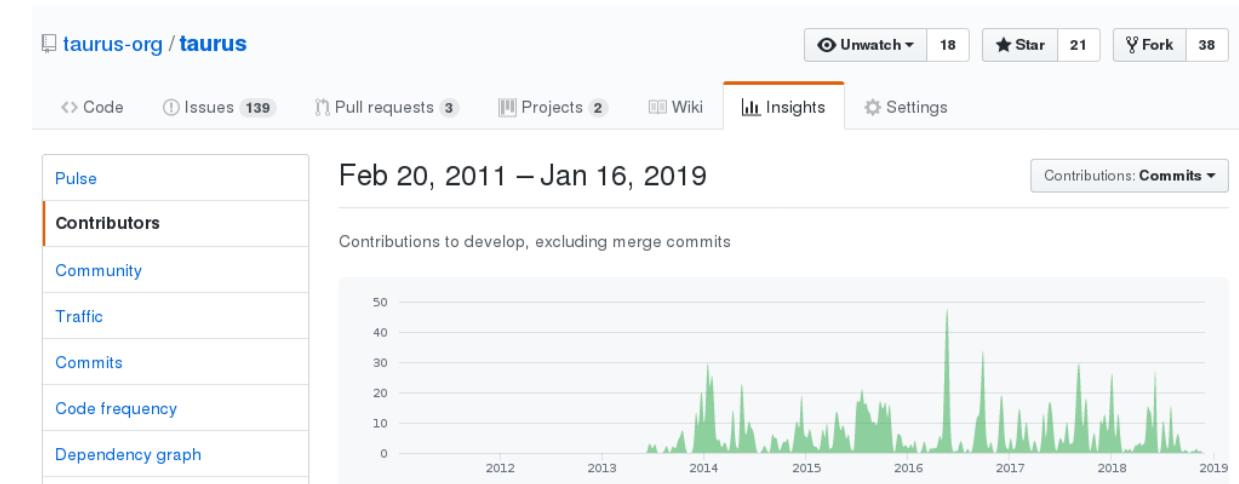
[sardana-org / sardana](#)

Code Issues Pull requests Projects Wiki Insights Settings

Feb 20, 2011 – Jan 16, 2019

Contributions to develop, excluding merge commits

Contributions: Commits



Community – Tools & Events



SEP TEP
 Sardana Enhancement Taurus Enhancement
 Proposal Proposal

SEP index: <https://sardana-controls.org/sep/index.html>
TEP index: <https://taurus-scada.org/tep/index.html>



Sardana Enhancement Proposals

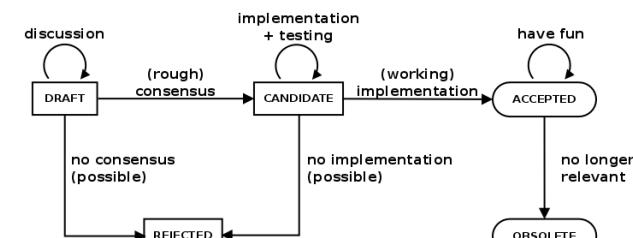
This is the main index of the Sardana Enhancement Proposals (SEP).

Each proposal should be in a separate file and be linked in the following table

Proposals list

Link	status	Title
SEP0	OBSOLETE	Introducing Sardana Enhancement Proposal
SEP1	OBSOLETE	Reorganization of code repos
i SEP2	CANDIDATE	Improve integration of 1D and 2D experimental channels
i SEP3	REJECTED (handled in #297)	Adapt to TEP3 (Tango-independent taurus.core)
i SEP4	ACCEPTED	HKL integration
i SEP5	ACCEPTED	Implementation of tests infrastructure
i SEP6	ACCEPTED	Continuous Scan Implementation
SEP7	OBSOLETE	Code contribution workflow
SEP8	CANDIDATE	Remove from Taurus objects t
SEP9	ACCEPTED	Compact Read+Write widget:
SEP10	OBSOLETE	Taurus separation
SEP11	ACCEPTED	Direct load of .ui files
SEP12	CANDIDATE	Use python Enum instead of t
SEP13	REJECTED (moved to TEP13)	Unified plugins support in Tau
SEP14	DRAFT	MSENV taurus schema
SEP15	ACCEPTED	Moving Sardana to Github
i SEP16	DRAFT	Plugins (controllers, macros, etc.) register
SEP17	DRAFT	Ongoing acquisition formalization and implementation
i SEP18	ACCEPTED	Extend acquisition and synchronization concepts for SEP2 needs

Proposal states



Community – Tools & Events



Videoconf: <https://meet.jit.si/sardana>
<https://github.com/sardana-org/sardana-followup>



sardana-org / sardana-followup

Code Issues 9 Pull requests 1 Projects 0 Wiki Insights Settings

Unwatch 9 Unstar 2 Fork 6

Minutes and other stuff gathered during the Sardana project follow-up meetings Edit

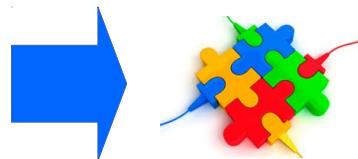
Manage topics

162 commits 3 branches 0 releases 8 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

reszelaz	Merge pull request #37 from reszelaz/20190110-minutes ...	Latest commit 4f3029a 5 days ago
20150610-DESY	Upload Talks DESY15 WS	7 months ago
20171102	Formatting changes	a year ago
20171211	Adding tasks	a year ago
20180125	Merge pull request #7 from amilan/2018_01_25_meeting	11 months ago
20180405	corrected minor mistakes	10 months ago
20180517	Alba presentation	4 months ago
20180605-Prague	Removing moved files	7 months ago
20180619	Add sardana docs presentation	7 months ago
20180710-Barcelona	Update DocsStyleGuide.md	6 months ago
20180913	Fix typos	4 months ago
20181004	update agenda to reflect points covered during meeting	3 months ago
20181108	Error from ctrl-C at DESY	2 months ago
20181213	Applying Zibi suggestions	a month ago
20190110	Add minutes	6 days ago
README.md	Initial commit	a year ago

Community – Tools & Events



Plugins Catalogue: <https://github.com/sardana-org/sardana-plugins>



sardana-org / sardana-plugins

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Branch: master [sardana-plugins / hardware.md](#)

 reszelaz Fix typo in hardware category 66e5a73 on Aug 8

1 contributor

14 lines (11 sloc) | 928 Bytes

[Raw](#) [Blame](#) [History](#)

Sardana plugins for specific hardware

Below you will find a table with Sardana plugins for specific hardware like for example motion controllers, detectors, etc.

Name	Description	Link(s) to project
ALBA Em Electrometer	Low current electrometer	sardana-albaem
AdLink	AdLink DAQ cards e.g. 2005	sardana-adlink
IcePAP	IcePAP motion controller	sardana-icepap
Linkam	Temperature controller	sardana-linkam
LoCuM	Low current monitor	sardana-locum
OPUS	Bruker OPUS spectrometer	sardana-opus

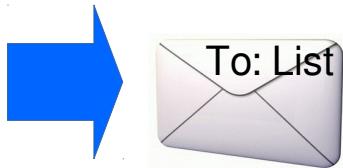
Community – Tools & Events



Training and workshops: <https://github.com/sardana-org/sardana-training>
(satellite to Tango meetings or ICALEPCS conference)



Community – Tools & Events



Announcements: sardana-users@lists.sourceforge.net
tauruslib-users@lists.sourceforge.net

Coordination: sardana-devel@lists.sourceforge.net
tauruslib-devel@lists.sourceforge.net

Questions?

... you can also reach us at any time!

Acknowledgements:

- DESY: Teresa Nuñez, Jan Kotański, Tim Schoft, Thorsten Kracht
- SOLARIS: Grzegorz Kowalski, Stanisław Cabała, Ireneusz Zadworny, Michał Fałowski
- MAXIV: Antonio Milan, Juliano Murari, Paul Bell, Antoine Dupré, Miriam Lindberg, Vincent Hardion
- SOLEIL: Frédéric Picca
- Max Born Institute: Daniel Schick
- ALBA: all the Computing Division