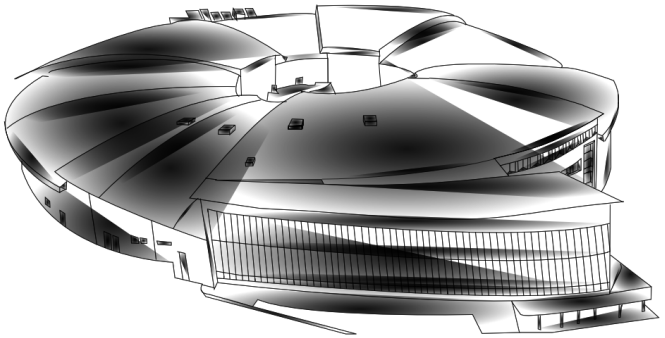
Taurus: Big & Small

from Particle Accelerators to Desktop Labs



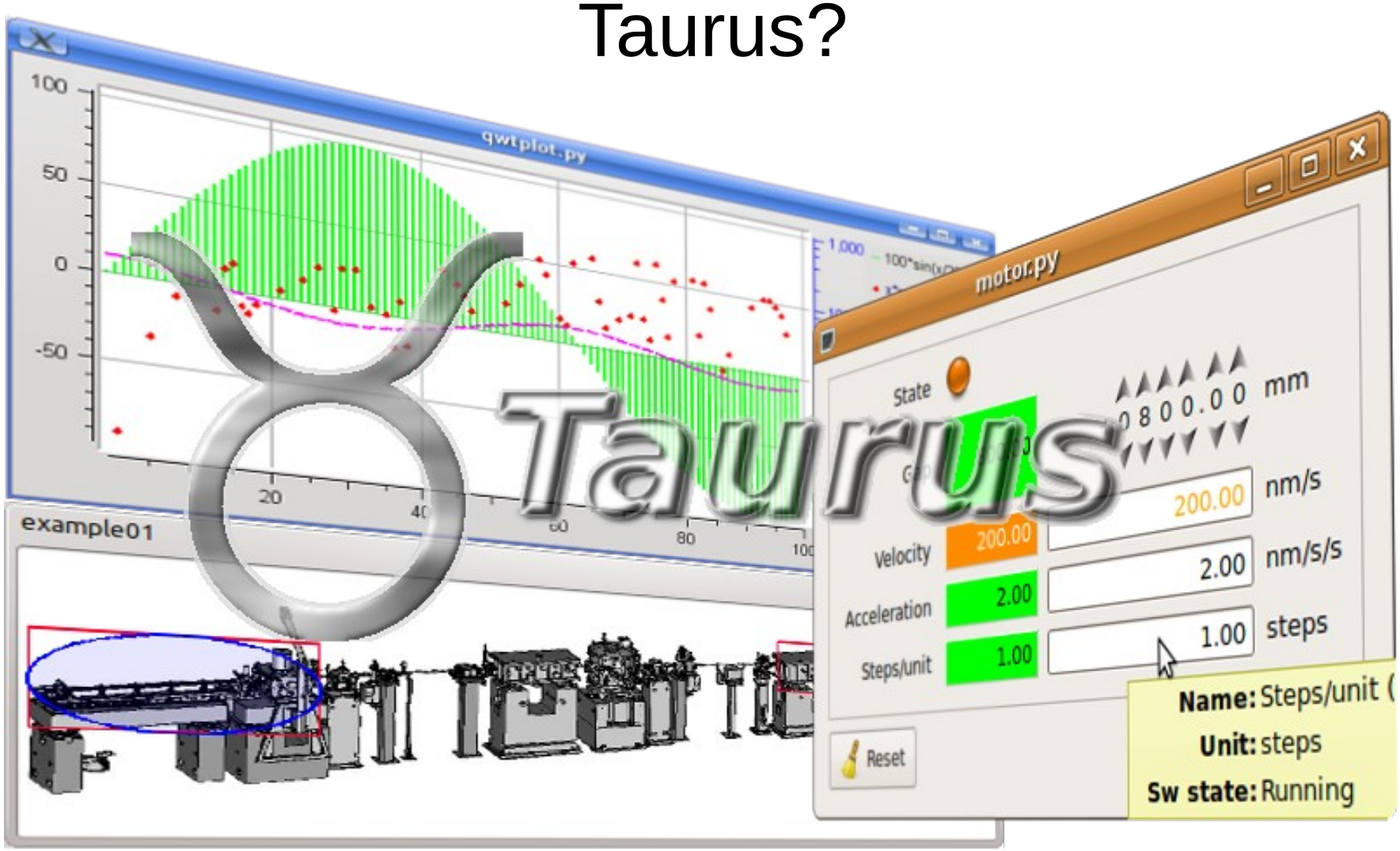
by: Carlos Pascual-Izarra

+ G. Cuní, C. Falcón-Torres, D. Fernández-Carreiras, Z. Reszela, M. Rosanes, Oscar Prades-Palacios

ICALEPCS2017

ALBA

Taurus?

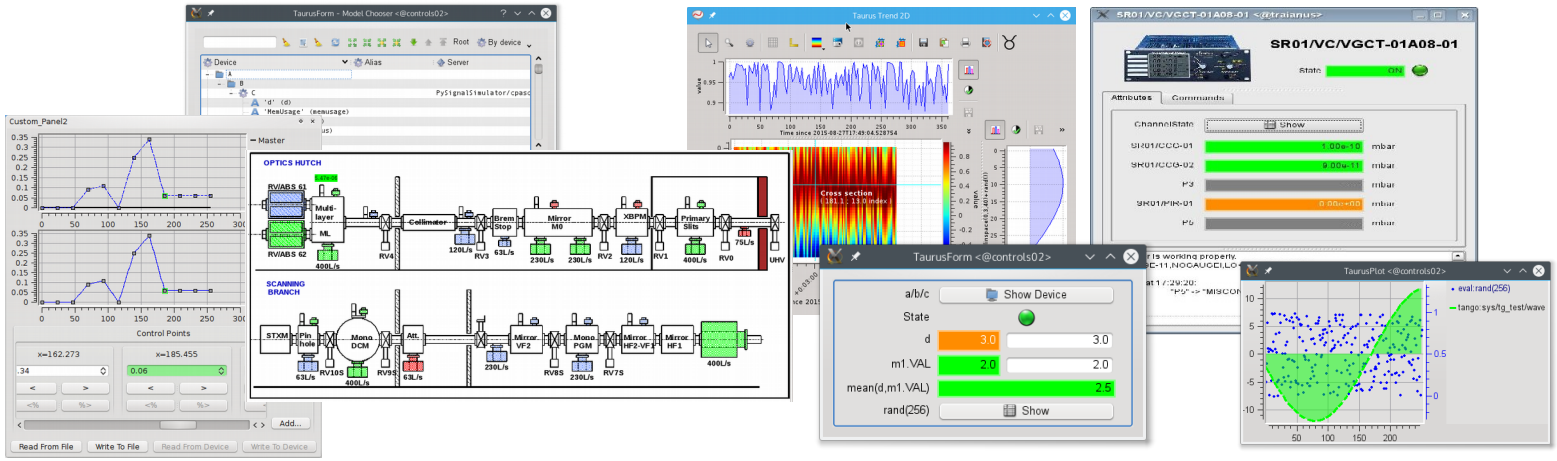


- **Taurus** is a framework for building control and data acquisition **CLIs** and **GUIs**
- It is based on **Python** and extends **PyQt**
- It supports plugins for various control systems (**Tango**, **EPICS**,...) or data sources (**HDF5**, **Python eval**,...)

MORE INFO https://taurus-scada.org

Structure of Taurus

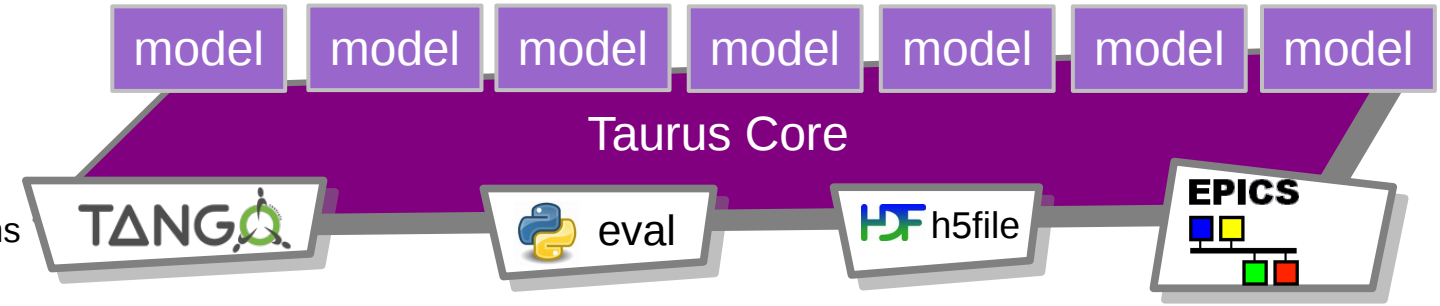
Taurus Qt Widgets



Taurus Core

Model Objects

Scheme plugins



Data Sources

Approaches to interface with unsupported sources

e.g., how to read data from a HDF5 file as an attribute?

Distributed Control System

Find (or write a new) DS capable of exposing contents of an hdf5 file as a Tango attribute...

h5file

Custom scheme

Use the custom h5file scheme plugin

```
h5file:/tmp/foo.h5::entry/time
```

eval

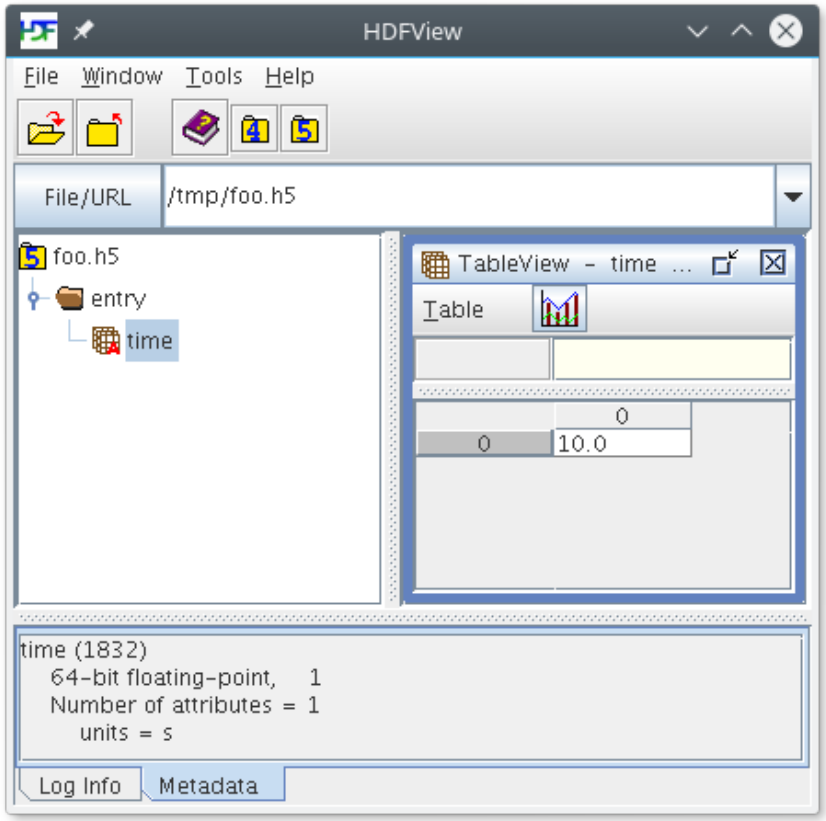
Evaluation scheme

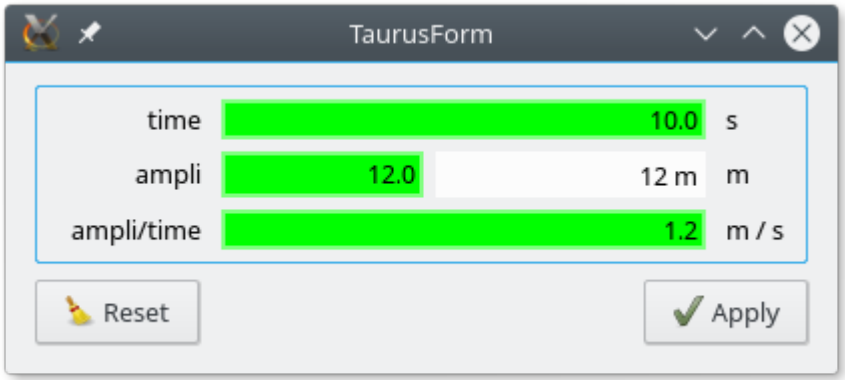
Use the eval scheme with the h5py module

```
eval:@f=h5py.File("/tmp/foo.h5")/f["entry/time"][:]
```

h5file scheme

```
$> pip install git+https://github.com/taurus-org/h5file-scheme.git
$> echo 'EXTRA_SCHEME_MODULES = ["h5file"]' >> taurus/tauruscustomsettings.py
$> taurusform h5file:/tmp/foo.h5::entry/time \
              tango:sys/tg_test/1/ampli \
              eval:{tango:sys/tg_test/1/ampli}/{h5file:/tmp/foo.h5::entry/time}
```





HDF5 TANGO eval

MORE INFO https://github.com/taurus-org/h5file-scheme

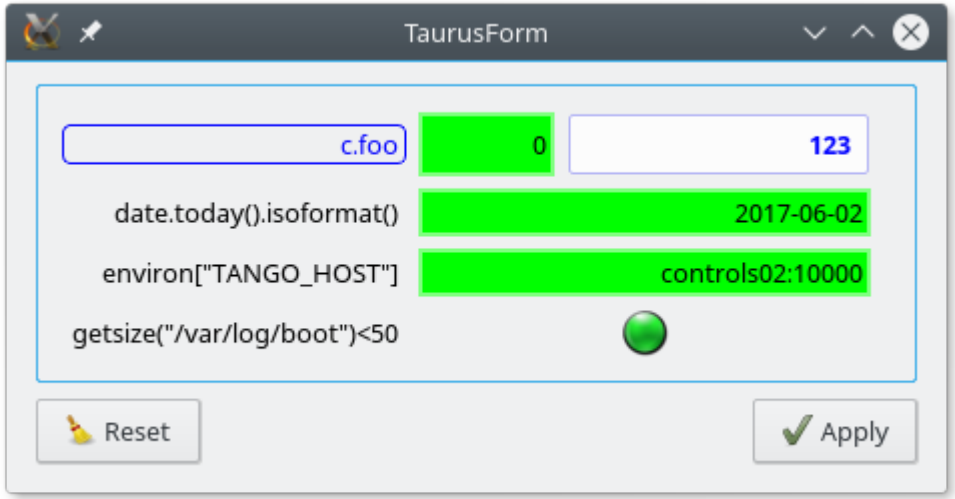
Eval scheme

- Allows mathematical evaluations
- Support writable eval attributes
- Use **any module or class** as a **custom evaluator**


```
$> taurusform 'eval:@c=mymod.MyClass()/c.foo'                    \
              'eval:@datetime.*/date.today().isoformat()'        \
              'eval:@os.*/environ["TANGO_HOST"]'                 \
              'eval:@os.path.*/getsize("/var/log/boot")<50'
```

mymod.py

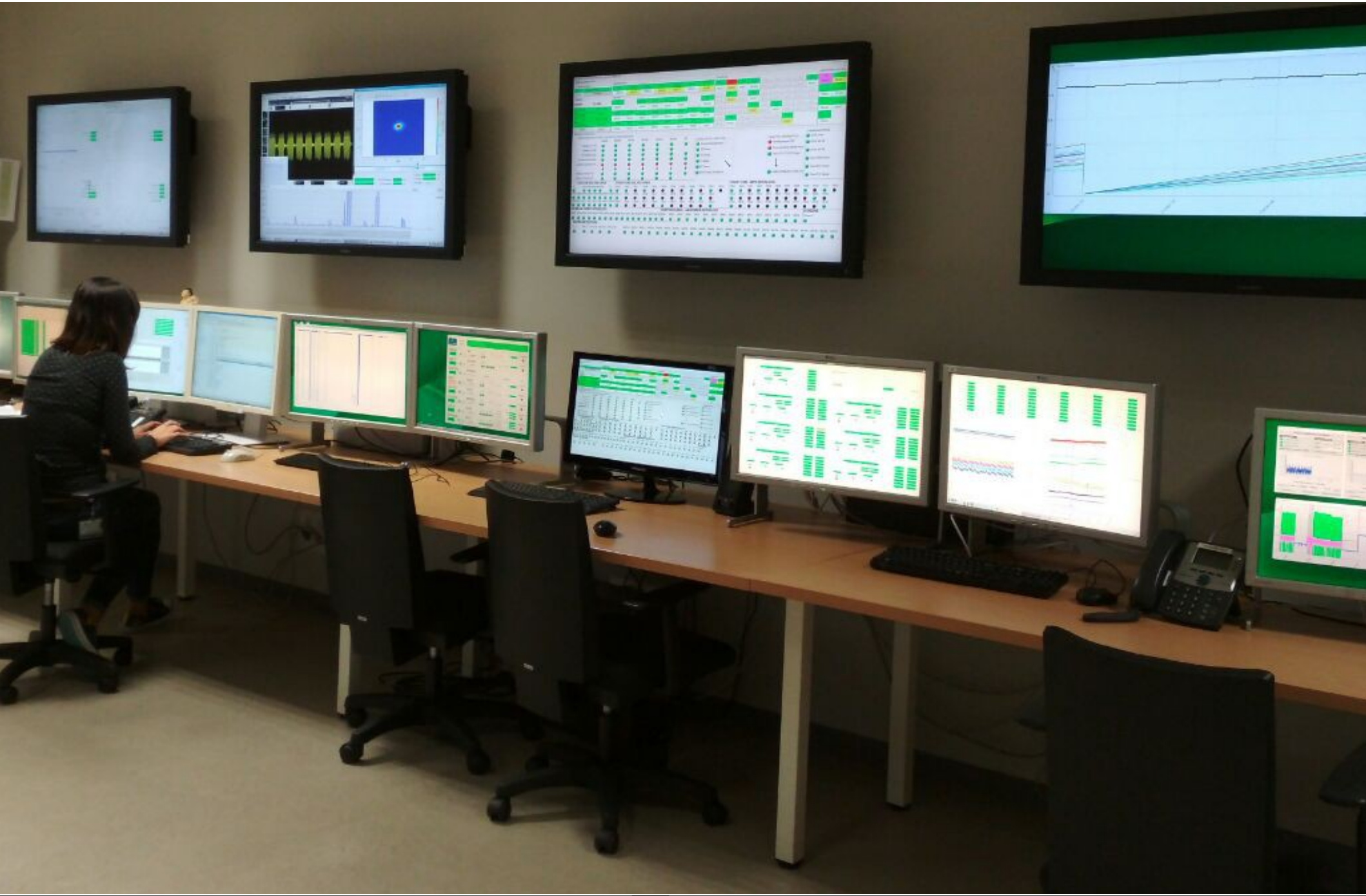
```
class MyClass(object):

    _foo = 0

    def get_foo(self):
        return self._foo

    def set_foo(self, value):
        self._foo = value

    foo = property(get_foo, set_foo)
```



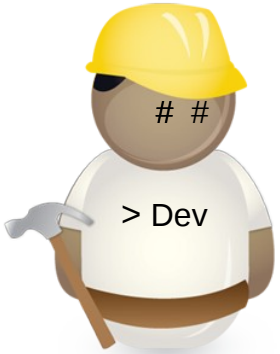
MORE INFO

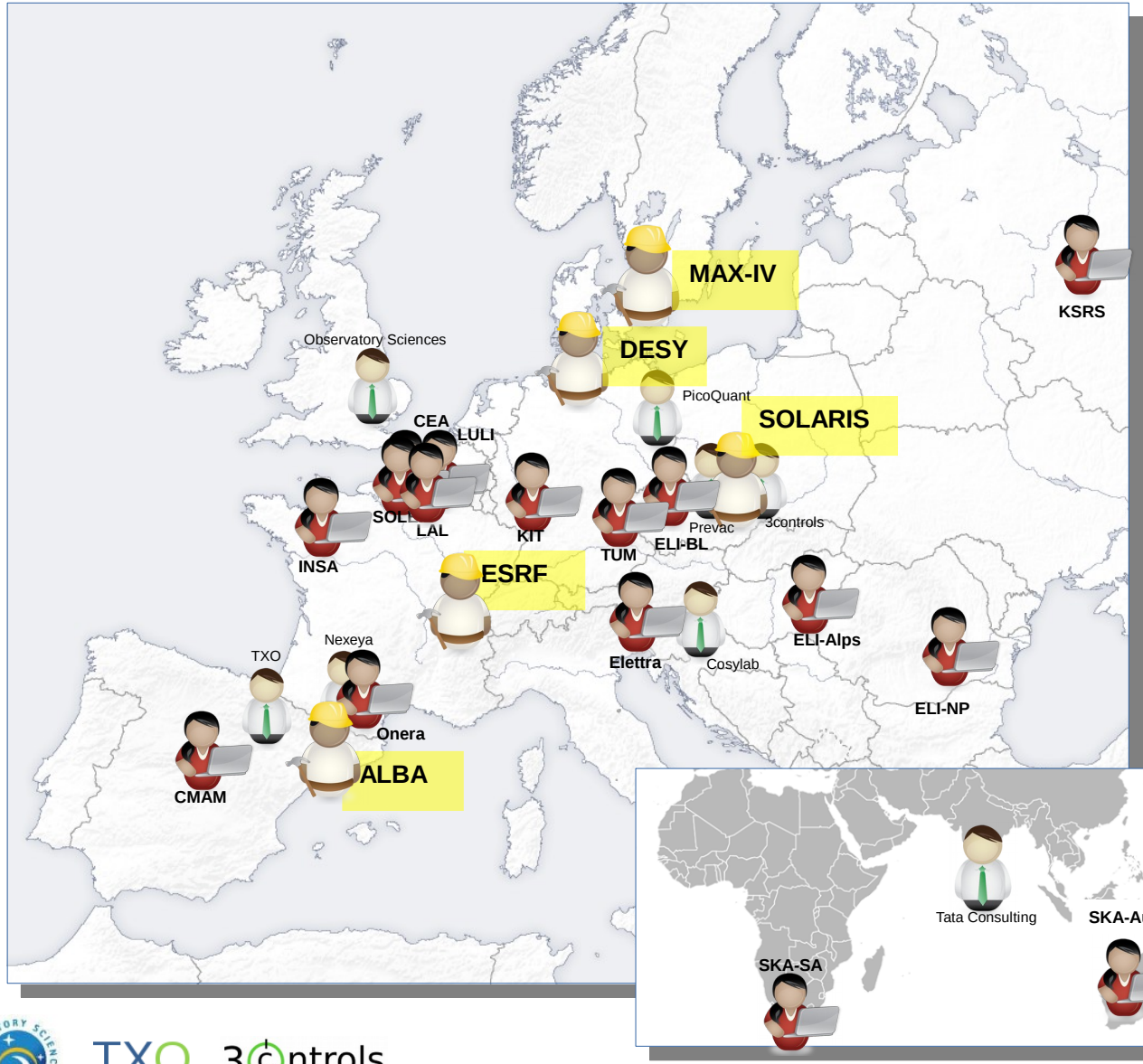
docs: http://www.taurus-scada.org/devel/api/taurus/core/evaluation.html
mymod example: taurus.core.evaluation.test.res.mymod

Taurus in large facilities



Taurus Community





ALBA

DESY

ESRF

MAX IV

SOLARIS
NATIONAL SYNCHROTRON RADIATION CENTRE

debian

NEXEYA

COSYLAB

OBSERVATORY SCIENCES

PICOQUANT

TXO

PREVAC

3controls

TATA
TATA CONSULTANCY SERVICES

TUM

KIT

Karlsruhe Institute of Technology

eli
extreme light infrastructure

cea

LAL

INSA
RENNES

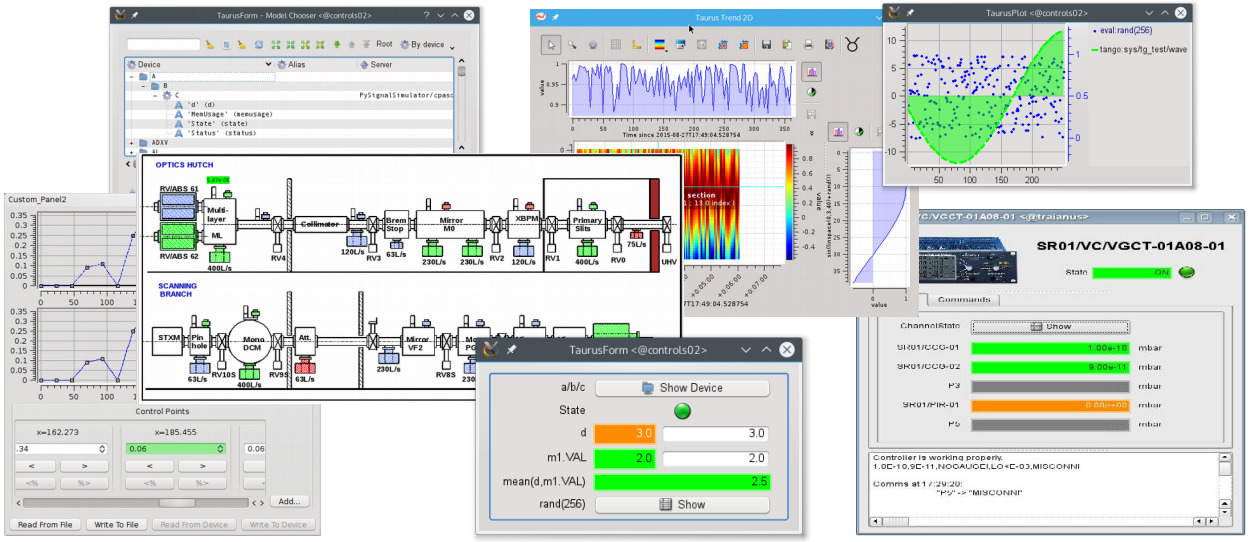
ONERA
THE FRENCH AEROSPACE LAB

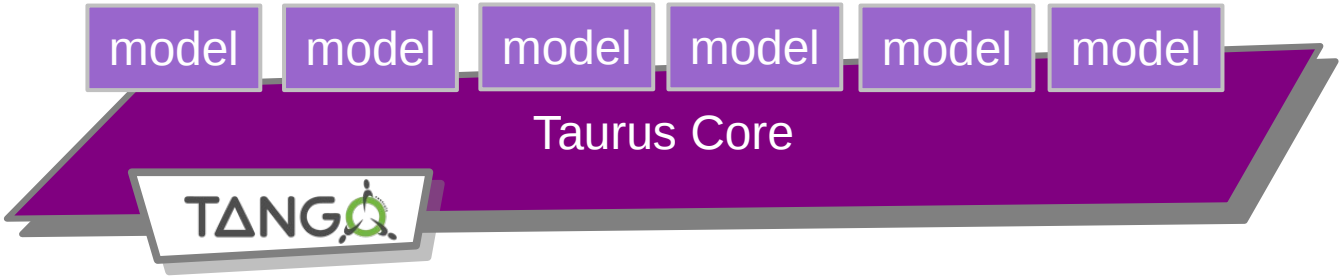
CMAM

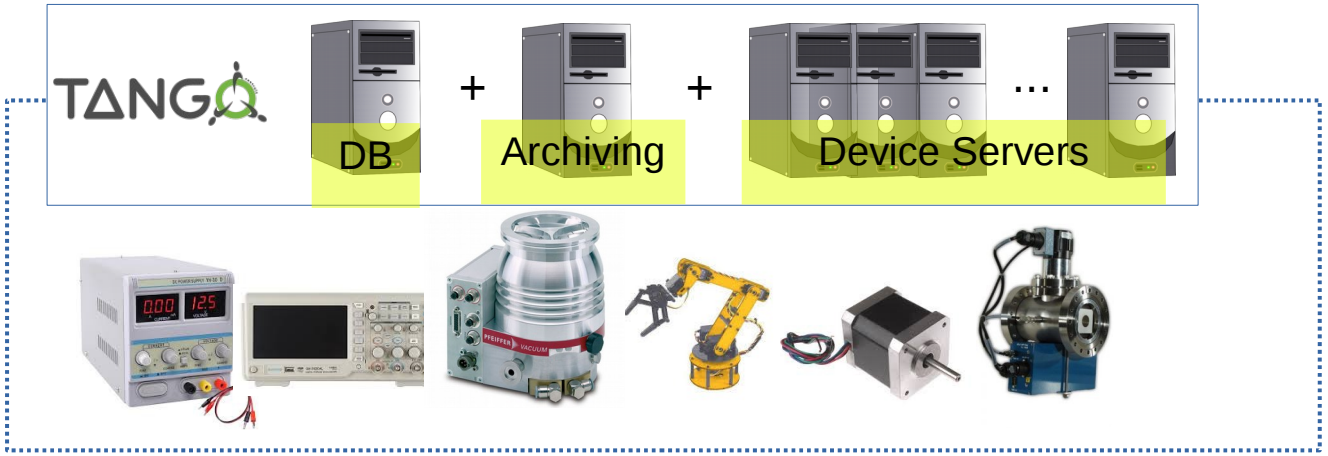
SKA
SQUARE KILOMETRE ARRAY

SOLEIL
SYNCHROTRON

Taurus in large Facilities: example of ALBA

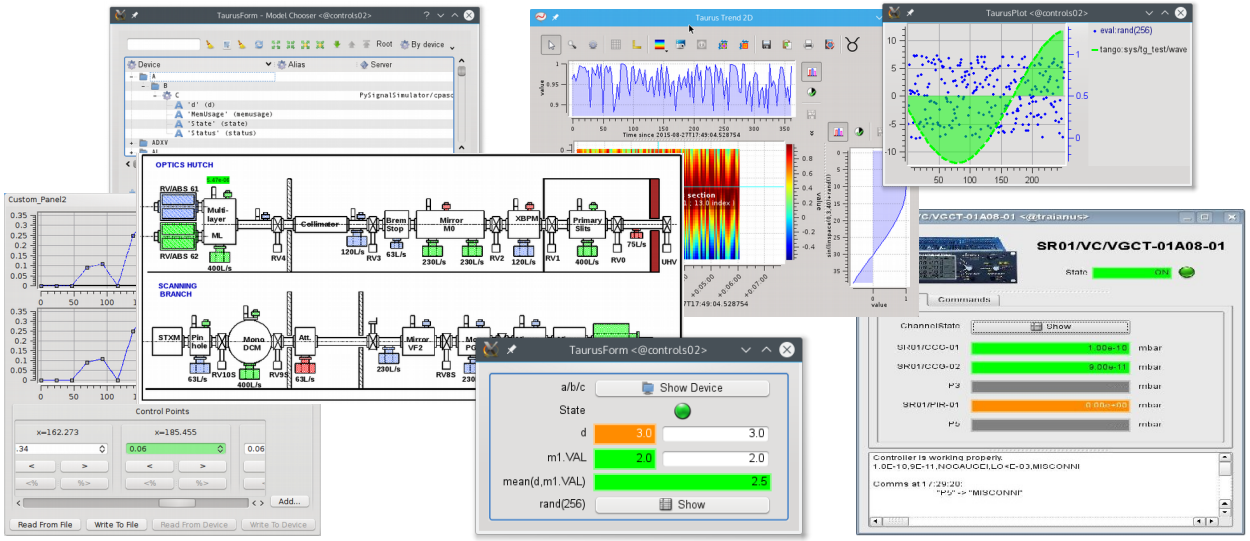


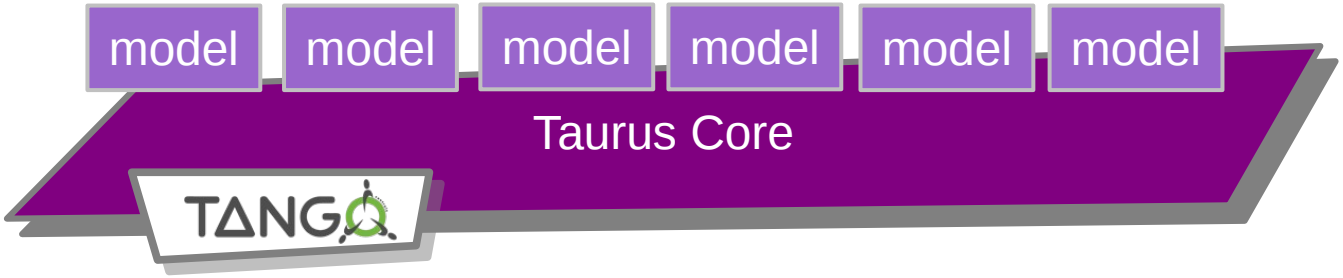


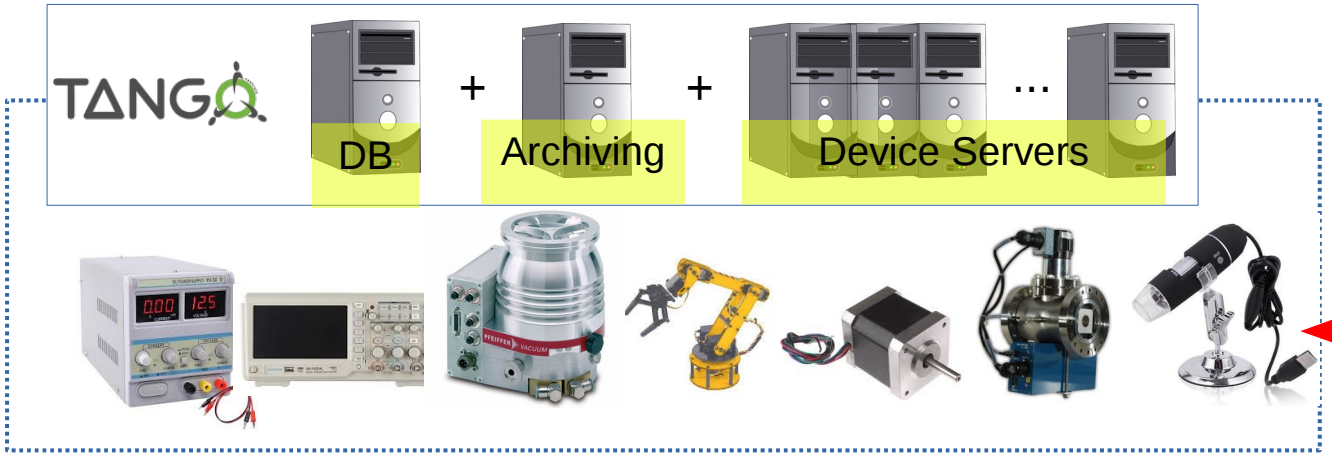


- Most of the control system relies on Tango
- ~100 Taurus GUIs
- ~300 machines
- ~10 Tango DBs
- ~100K Tango attributes
- New hardware **?**

Taurus in large Facilities: example of ALBA

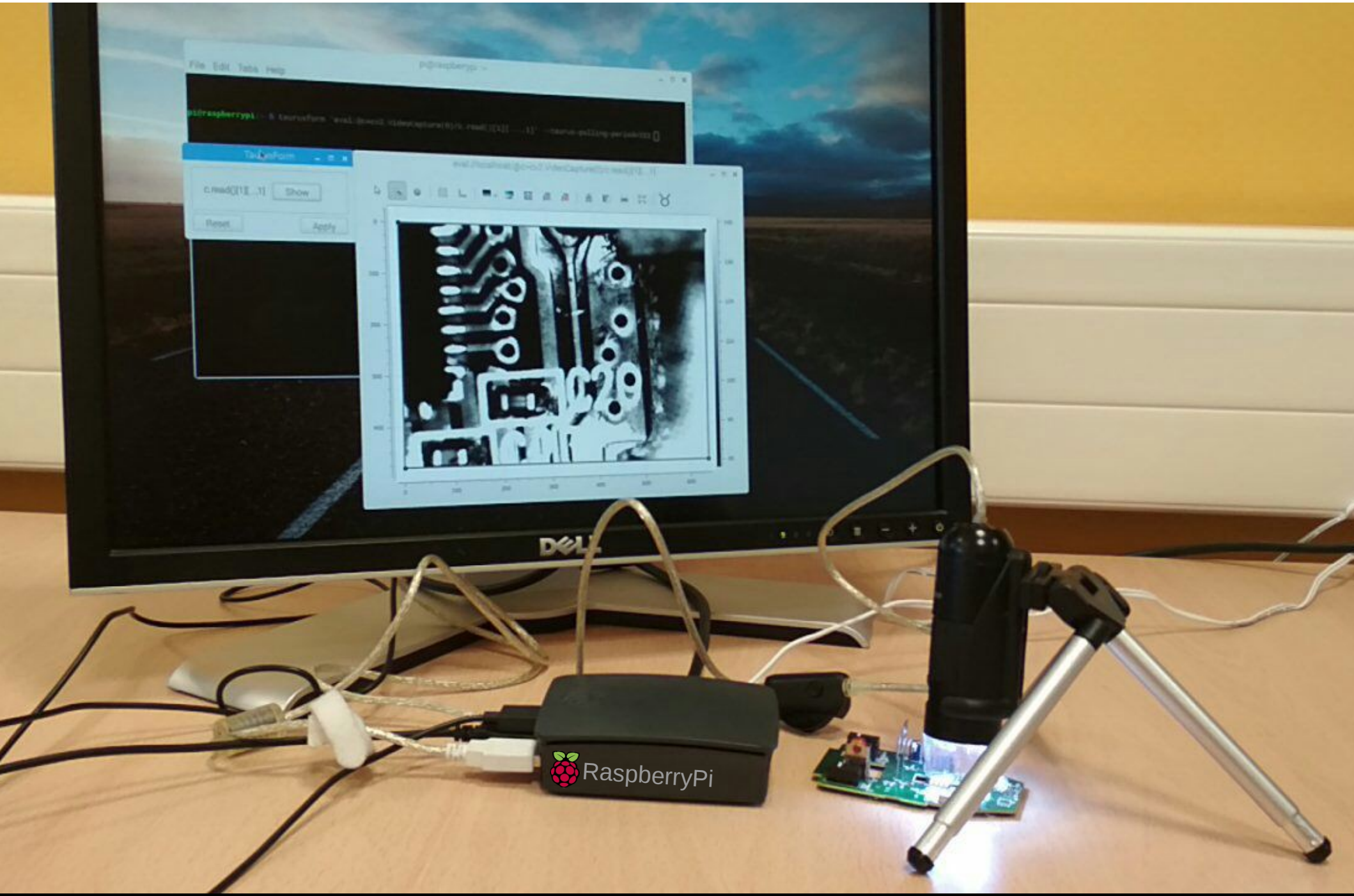




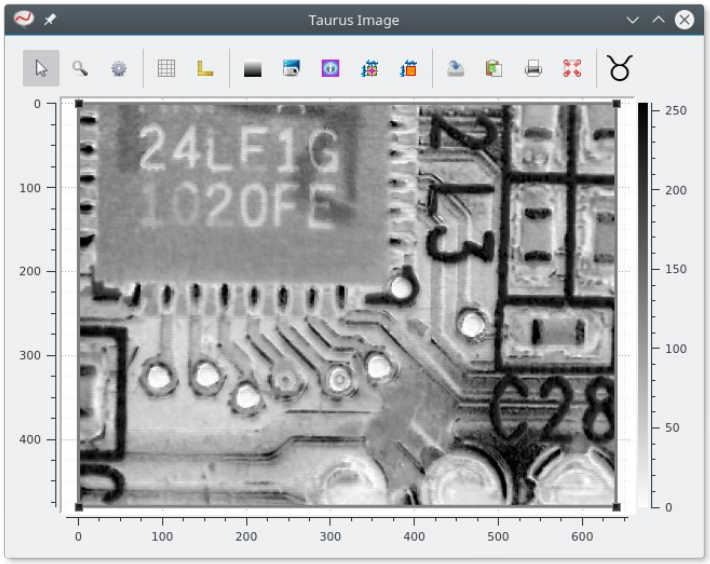


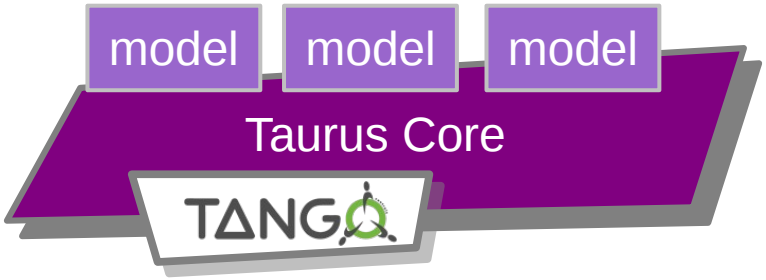
- Most of the control system relies on Tango
- ~100 Taurus GUIs
- ~300 machines
- ~10 Tango DBs
- ~100K Tango attributes
- New hardware → write a Tango Device Server

Taurus in “Desktop Labs”:



Taurus in “Desktop Labs”





Approach 1

Use same tools as for large facilities (scale-down): use a Distributed Control System on a single machine

- Install Taurus + Tango on a single machine:

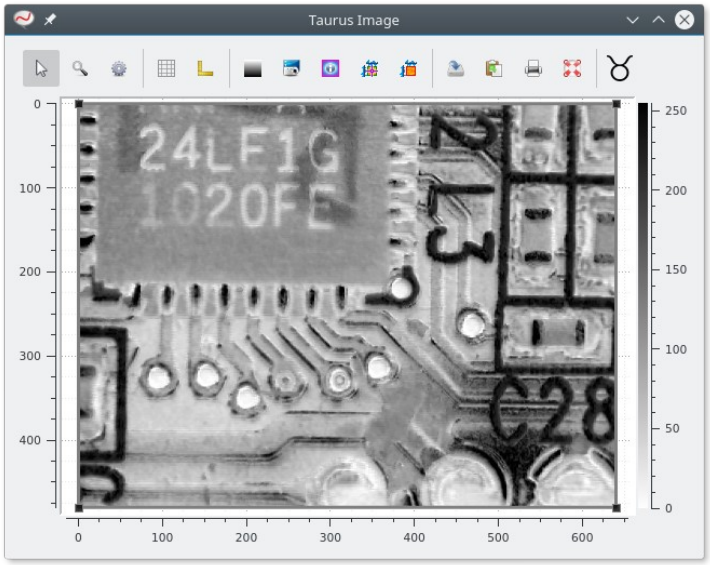
```
$> apt-get install default-mysql-server
$> apt-get install tango-db
$> apt-get install python-taurus
```

- Install, configure and run a Device Server supporting your hardware (or write a new one)

Recommended **if**:

- you are already familiar with Tango/EPICS
- the hardware is already supported by Tango/EPICS
- you do not mind the communication overhead

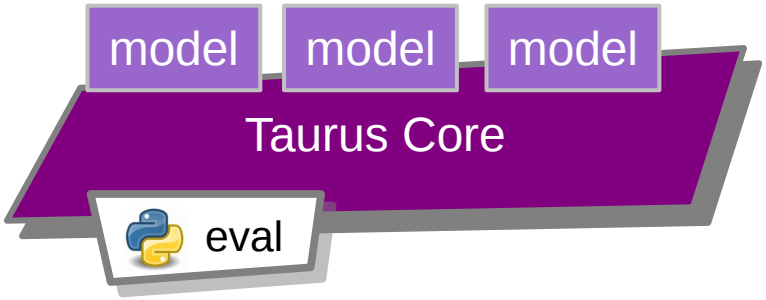
Taurus in “Desktop Labs”



Approach 2

Use the eval scheme and connect directly

```
$> apt-get install python-taurus

$> taurusimage 'eval:@c=cv2.VideoCapture(0)/c.read()[1][...,1]'
```



Recommended **for**:

- single-machine systems
- quick prototyping
- quick support of new hardware

https://taurus-scada.org

Welcome to Taurus Ho

taurus-scada.org

taurus

latest

Search docs

Home Page

Project Page

Download from PyPI

Taurus 4.1 documentation

Read the Docs v: latest

Docs » Welcome to Taurus Home Page!

Edit on GitHub

Welcome to Taurus Home Page!