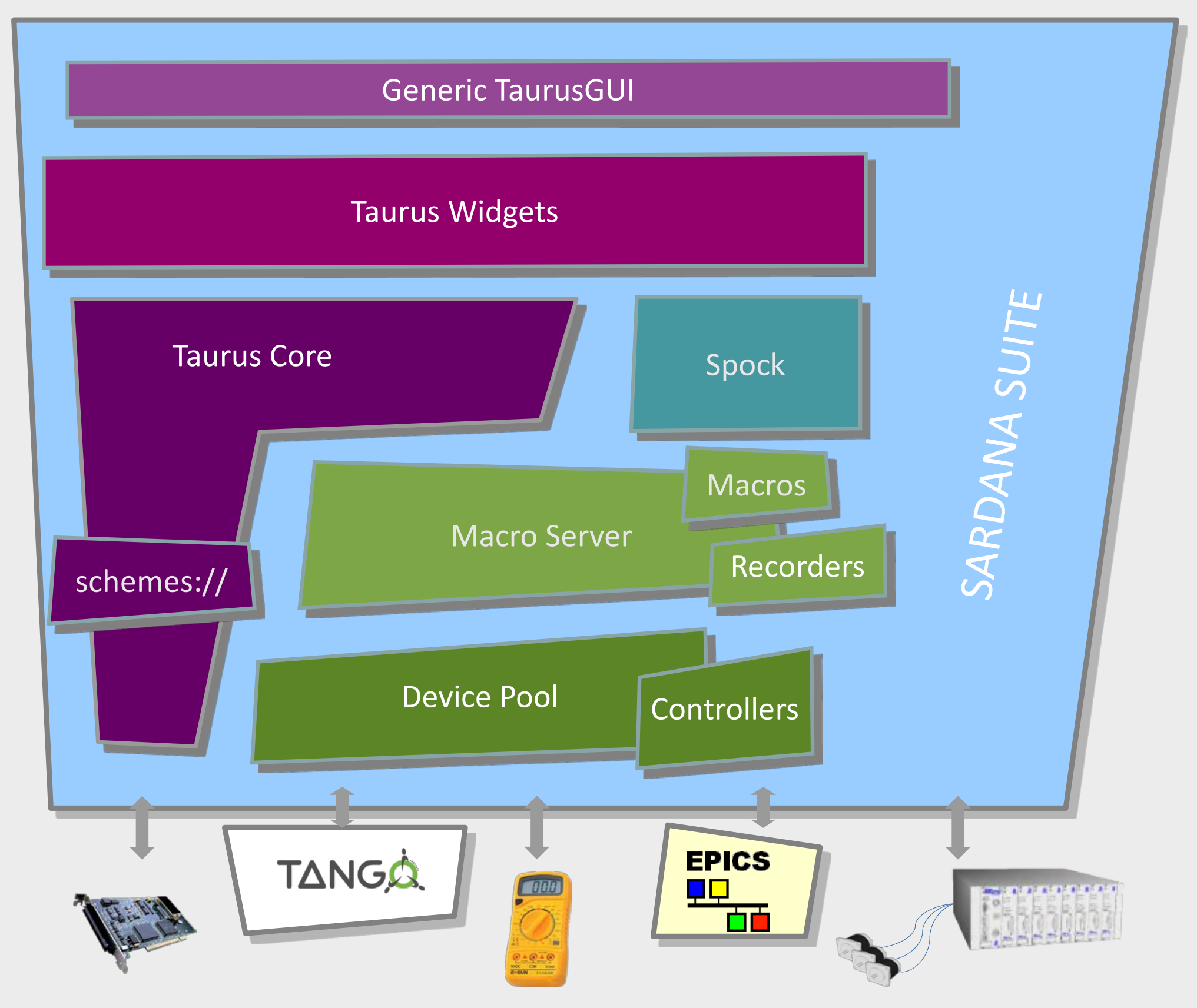
THPDP050

Improving User Experience and Performance in Sardana and Taurus: A Status Report and Roadmap

Z. Reszela, J. Aguilar, M. Caixal i Joaniquet, G. Cuní, R. Homs-Puron, E. Morales, M. Navarro, C. Pascual-Izarra (on leave), J. Ramos, S. Rubio-Manrique, O. Vallcorba (ALBA-CELLS Synchrotron, Barcelona, Spain),
M. T. Nunez Pardo de Vera (DESY, Hamburg, Germany), B. Bertrand, J. Forsberg (MAXIV Laboratory, Lund, Sweden), M. Piekarski (Solaris, Krakow, Poland)
D. Schick (MBI Berlin, Berlin, Germany)

SARDANA – Scientific SCADA Suite

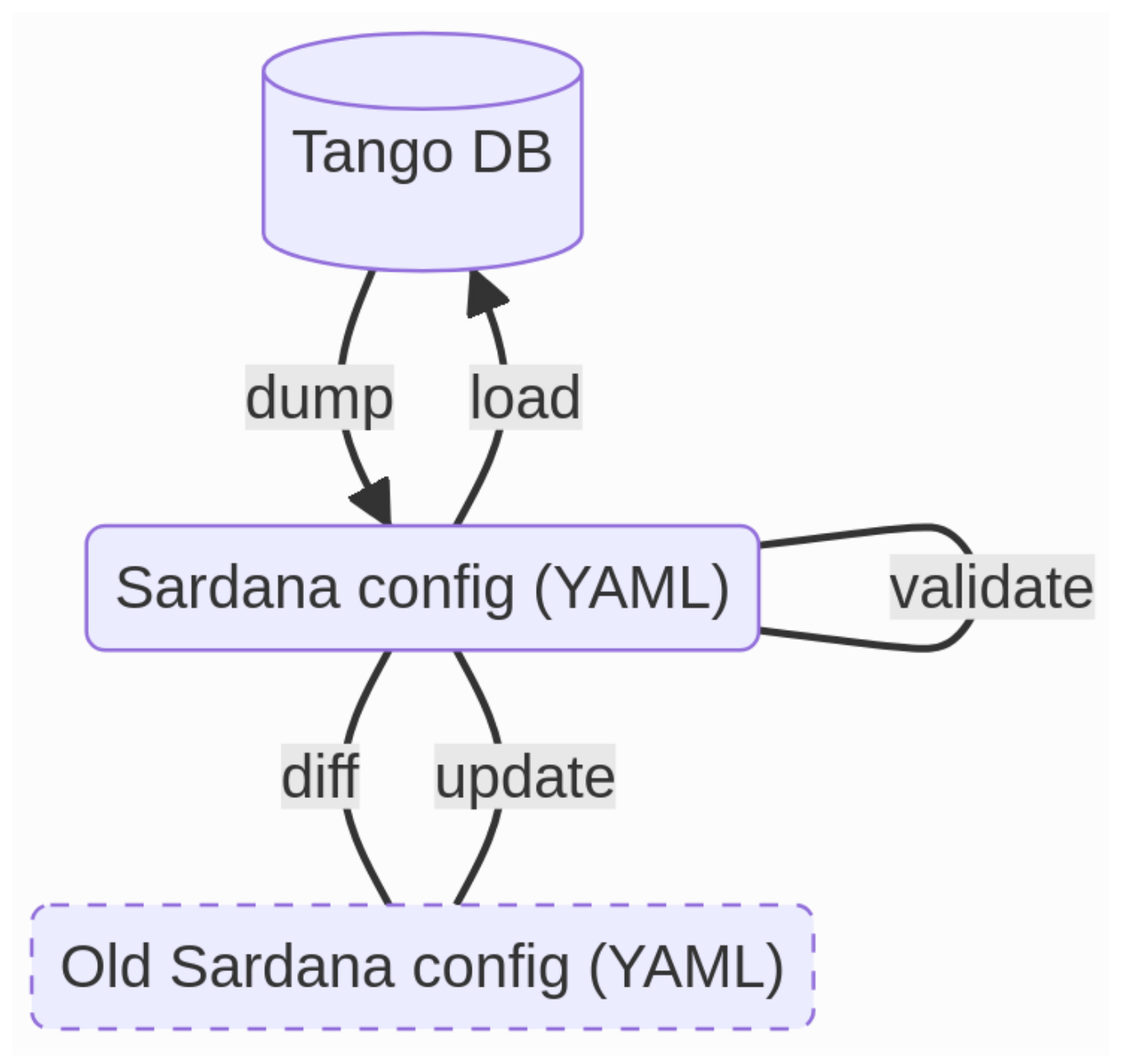


- Community-driven
- Widely used
- Production-ready
- Well supported
- Actively developed
- Free/Open Source
- Modular
- Multi-platform
- Easy to install
- Based on Python
- Built on top of TANGO

DEVELOPER EXPERIENCE

Configuration

- New YAML based format describes the whole Sardana system.
- CLI tools for load, dump, etc.
- Deleting Sardana elements at runtime now checks the existence of dependent elements.



Sardana CLI configuration tools

Programming interface

- Trigger/Gate Controller now supports multiplexor mode and gets synchronization description in *dial position units*.
- New Taurus multi-model API simplifies development of model *container* and model *composer* widgets.

```
class MyTriGaCtrl(TriggerGateController):

    [...]

    def SetAxisPar(self, axis, par, value):
        # set axis parameter: active_input
        [...]

    def SynchOne(self, axis, description):
        # description in dial position units

    [...]
```

Skeleton of the trigger/gate controller

Testing

- PyTest fixtures can be used for Sardana core and plugins testing without the need to worry about setup and teardown.

```
import pytest
from sardana import State

ctrl_mark = pytest.mark.kwargs({
    "motctrl01": {
        "klass": "MyMotorController",
        "library": "MyMotorCtrl.py"
     }
})

pytestmark = [ctrl_mark]

def test_get_state(mot01):
    assert mot01.state == State.On
```

Test of controllers StateOne() method

Continuous Integration

- Docker images are now based on micromamba: support multiple python versions, are more modular, occupy less space and are easier to extend.
- CI now uses GitLab services.

Packaging

- PyPI and conda packages were split in several sub-packages.
- Now users can install only those dependencies which are needed.

Maintenance

- Both Sardana and Taurus code have been migrated to Python3.
- Taurus plotting widgets have been migrated from PyQwt5 to pyqtgraph.

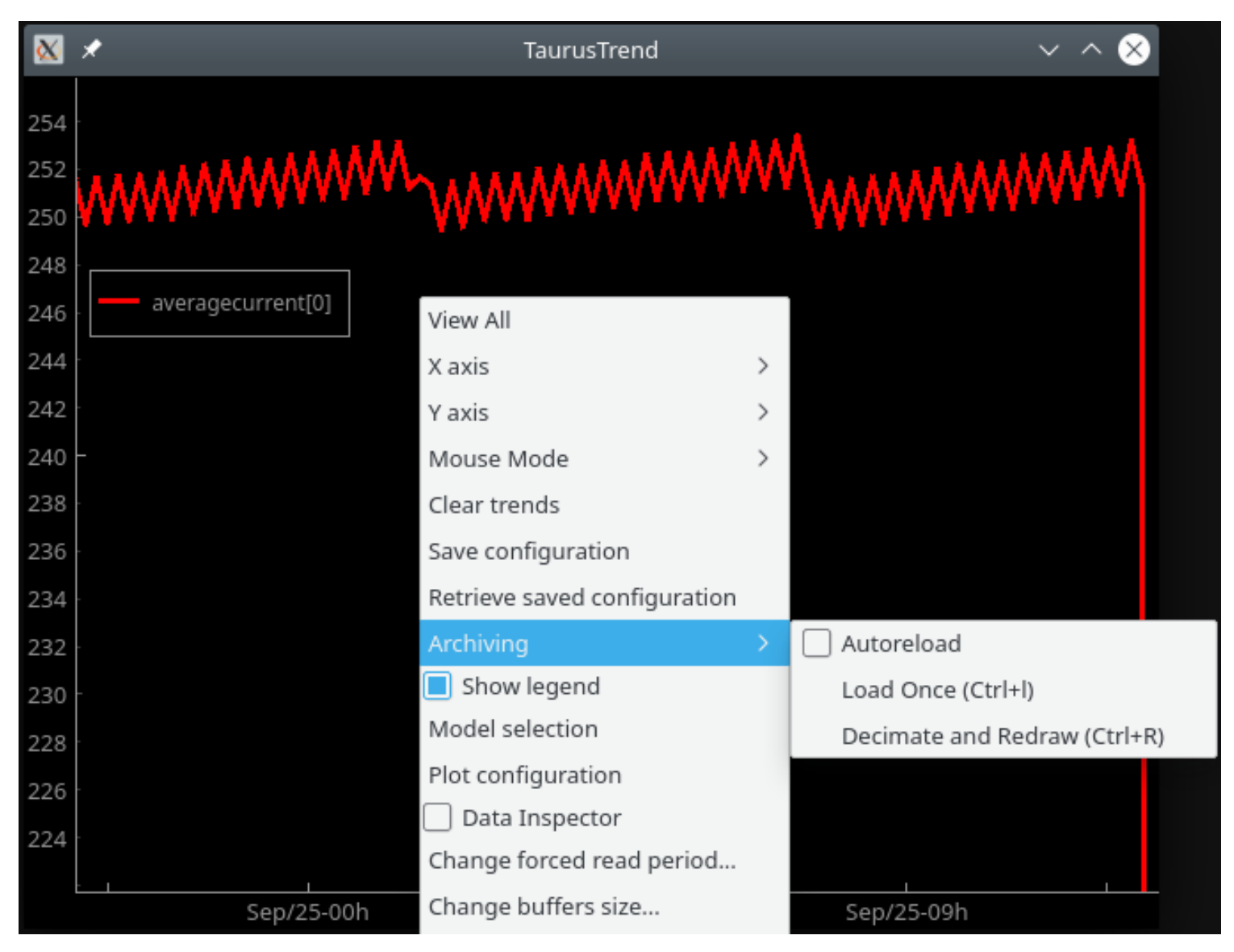
Type annotations

- Sardana automatically converted type information from docstrings into Python type hints.

USER EXPERIENCE

Trends with Live and Archived Data

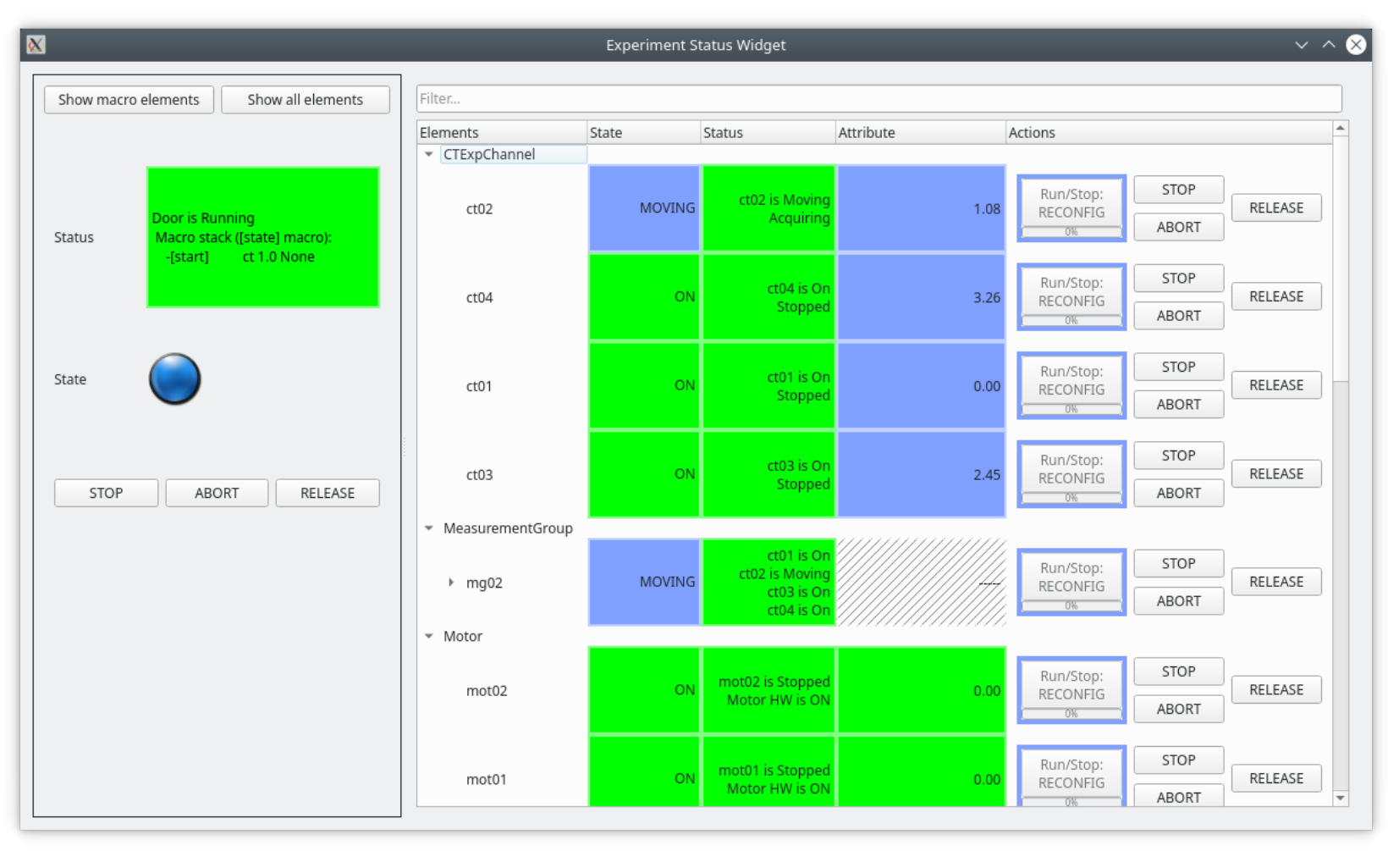
- Historical data is more accessible and insightful for users.
- At the same time the trend is continuously updated with new data being added.



TaurusTrend Widget showing archiving and live data

Debugging and Recovery

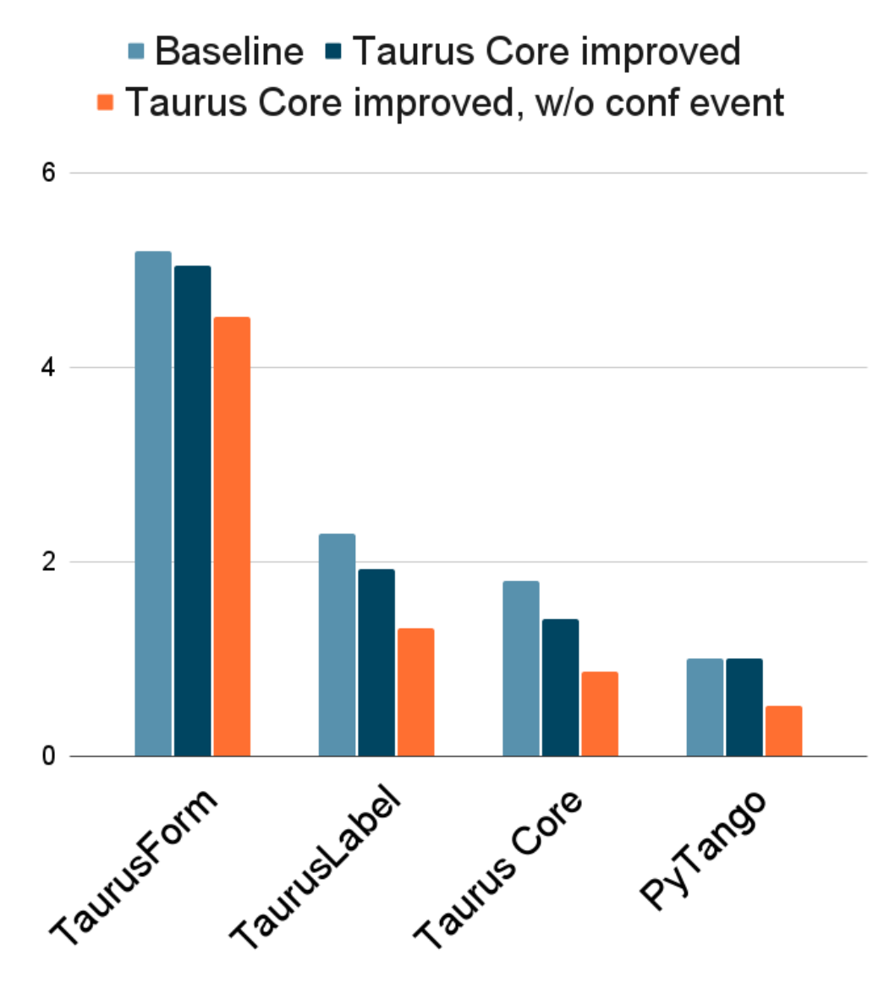
- GUI with real-time insights into the experiment status.
- Recovery of faulty elements with the *reconfig* functionality.



Experiment Status Widget

Taurus Performance Optimization

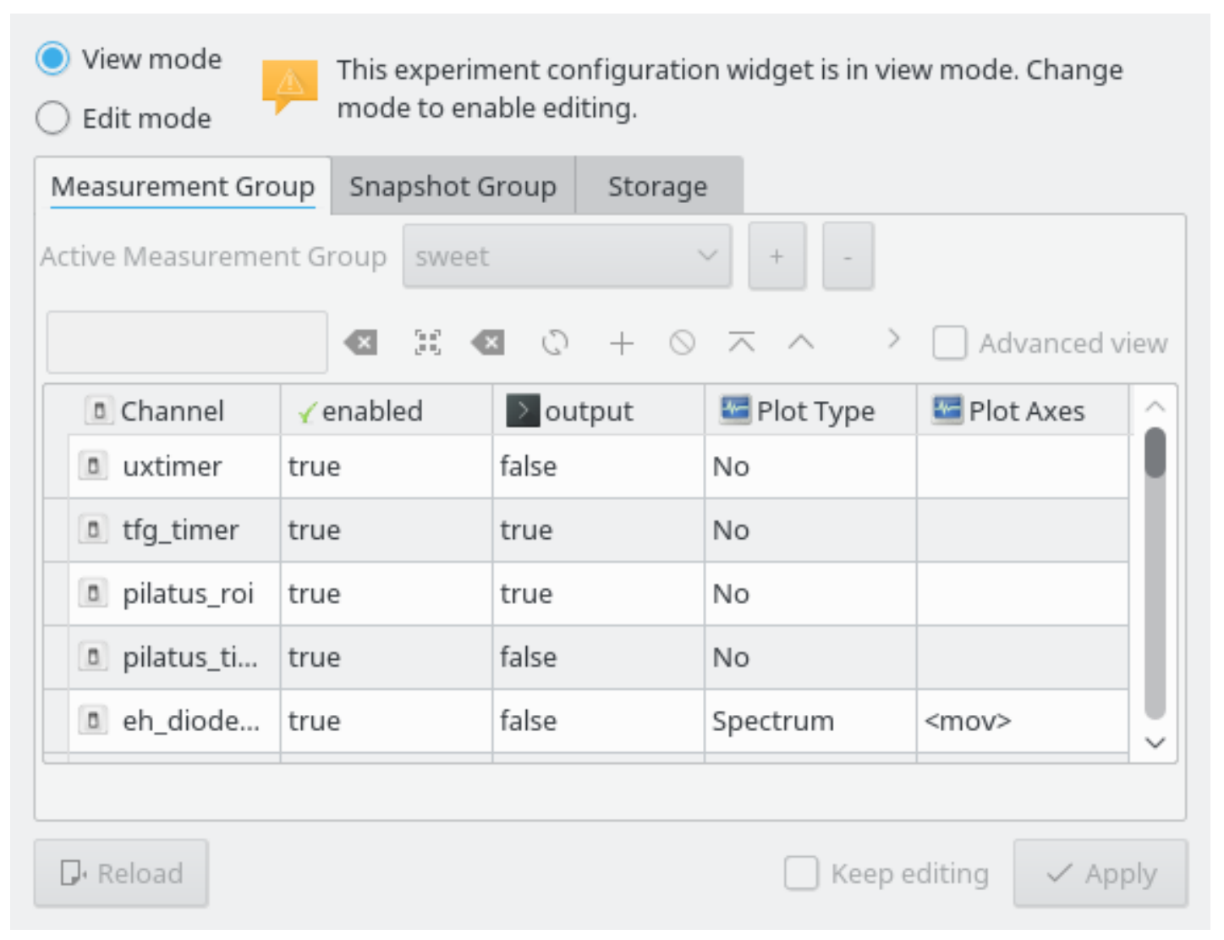
- Taurus Core's performance slows startup.
- Time profiling identified most critical issues.
- Optimization project is ongoing. Already reduced startup time up to ~50%.



Baseline measurement of relative startup times

Experiment Configuration

- Refined simple and advanced views.
- Introduced view and edit modes.
- New CLI version available.



This experiment configuration widget is in view mode. Change mode to enable editing.

Channel	enabled	output	Plot Type	Plot Axes
uxtimer	true	false	No	
tfg_timer	true	true	No	
pilatus_roi	true	true	No	
pilatus_ti...	true	false	No	
eh_diode...	true	false	Spectrum	<mov>

Experiment Configuration Widget

Documentation

- Added "What's new?" – CHANGELOG written in user friendly language.
- Started embedding video demos with short tutorials.

CLI

- Clearer and more accessible central point of access with single script with sub-commands.

ROADMAP

- Configuration: add support to environment and multiple files.
- Continuous scans:
  - Refactor Macro API for more flexible development of scans.
  - Add support for multiple synchronization descriptions.
  - Publication of data to an in-memory database.
- Reorganize documentation and provide tutorials for users.
- Further reliability and recovery improvements.
- Adapt Taurus to PyQt6.

MORE INFO

- `https://sardana-controls.org`
- `https://taurus-scada.org`

Acronyms and abbreviations
API – Application Programming Interface **GUI** – Graphical User Interface, **CLI** – Command Line Interface

ALBA SYNCHROTRON

www.albasynchrotron.es

ALBA Synchrotron
Carrer de la Llum 2-26
08290 Cerdanyola del Vallès,
Barcelona, Spain
Tel: +34 93 592 43 00