ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-WEMG017

MODULAR SCIENTIFIC SCADA SUITE WITH SARDANA AND TAURUS – LATEST DEVELOPMENTS

O. Vallcorba*, J. Aguilar, R. Homs-Puron, E. Morales, M. Navarro (on-leave),
J. Ramos, J. Moldes, F. Becheri, S. Rubio, Z. Reszela, ALBA-CELLS, Barcelona, Spain
M. T. Núñez Pardo de Vera, DESY, Hamburg, Germany
B. Bertrand, J. Forsberg, V. Da Silva, M. Lindberg, MAX IV Laboratory, Lund, Sweden
M. Piekarski, SOLARIS, Krakow, Poland
W. Kitka, S2 Innovation, Krakow, Poland
A. Hoffstadt, ESO, Garching, Germany
T. Braun, byte physics, Berlin, Germany
D. Schick, M. Schneider, MBI-Berlin, Germany

Abstract

Sardana and Taurus are community-driven, open-source SCADA solutions that have been used for over a decade in scientific facilities, including synchrotrons (ALBA, DESY, MAX IV, SOLARIS) and laser laboratories (MBI-Berlin).

Taurus is a Python framework for building both graphical and command-line user interfaces that support multiple control systems or data sources. Sardana is an experiment orchestration tool that provides a high-level hardware abstraction and a sequence engine. It follows a client-server architecture built on top of the TANGO control system.

In the last two years, significant developments have been made in both projects. Sardana focused on enhancing continuous scans, introducing multiple synchronization descriptions to support passive elements (e.g. shutters) and detectors reporting at different rates. The configuration tool has also been extended, following the roadmap defined by the community. Taurus has seen substantial performance gains, particularly in GUI startup times, as part of an optimization effort that started nearly three years ago. Latest improvements take profit of new TANGO event subscription asynchronous modes. Continuous codebase modernization is underway, and support for Qt6 is planned for the July 2025 release.

This paper reviews recent advances in Sardana and Taurus and outlines the current development roadmap.

SYSTEM OVERVIEW

Supervisory Control and Data Acquisition (SCADA) systems are essential in scientific facilities, providing the abstraction layers needed to integrate heterogeneous hardware, orchestrate experiments and offer usable interfaces.

Sardana [1, 2] and Taurus [3, 4] are Python-based, community-driven and open-source solutions currently used in production at several facilities. The main adopters of the Sardana–Taurus stack include ALBA, DESY, MAX IV, and SOLARIS synchrotrons and the MBI-Berlin and HI-JENA laser laboratories. Taurus is also used at the ESO observatory and, to a lesser extent, at ESRF, SOLEIL and ELETTRA synchrotrons. Together, Sardana and Taurus provide a modular and extensible SCADA suite covering the range from user interaction to low-level hardware orchestration:

- User interfaces: Taurus provides tools to create graphical interfaces (GUIs) that expose experimental setups, live data and control elements across multiple control systems through a unified abstraction model. Sardana adds extensions to Taurus to enhance the interaction with its specific components and in addition, it offers *Spock*, an IPython-based command-line interface (CLI).
- Experiment orchestration: Sardana includes a sequencing engine and high-level abstractions to coordinate devices. The *MacroServer* allows complex experimental procedures to be expressed as *macros*, while the *Device Pool* defines generic interfaces for motors, detectors and other equipment. This layer synchronizes elements and manages scans and data acquisition.
- Control system: Sardana is built on top of TANGO [5], a distributed client–server middleware widely used in large scientific facilities. TANGO provides standardized access to hardware and software devices and acts as Sardana's communication backbone.
- Hardware: Physical devices such as motors, detectors, shutters and sample environments exposed through the control system (e.g. TANGO devices).

This paper updates the ICALEPCS 2023 status report [6] and is organised from higher-level user interfaces (Taurus) down to orchestration layers (Sardana), followed by the roadmap agreed by the community.

USER INTERFACES

Taurus uses Qt [7] to build GUIs, supporting both PySide [8] and PyQt [9] bindings via the `taurus.external.qt` module. For plotting live and archive data, the `taurus_pyqtgraph` extension [10] provides 1D widgets, while 2D support is offered through `guiqwt`-based widgets [11].

* ovallcorba@cells.es

ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-WEMG017

Recently, significant effort has been dedicated to improve Taurus performance and modernize its codebase.

Performance Optimization

Certain Taurus GUIs suffered from long startup times when handling many attributes. To address this, the Taurus Performance Optimization (TPO) project was initiated, based on systematic profiling to identify bottlenecks across different layers of the Taurus stack. First results were already presented on the previous paper [6].

Phase 1 focused on general optimizations and improvements of the polling mechanism, used when attributes do not push change events. It fixed repeated TANGO Authority instantiation, eliminated unnecessary and empty polling threads, cached fully qualified device/attribute names and optimized `TangoAttribute::read()` caching. Taurus no longer forces an immediate read if event subscription fails, and values are populated on the first polling cycle, allowing the GUI to appear earlier. These changes reduced startup times by 27-44% (see Fig. 1, TPO columns) and are available from Taurus 5.2.

Phase 2 targets attributes pushing change events and leverages new TANGO event subscription modes that allow asynchronous subscription with optional entity reading [12]. The first implementation in Taurus evaluates (i) synchronous subscription while delegating the first read to Taurus in an independent thread, and (ii) the `AsyncRead` mode, where subscription and initial read are performed asynchronously at the TANGO level, and Taurus relies on `EventReason` in the first `EventData` to enable or disable polling accordingly. Early benchmarks bring startup times closer to the PyTango baseline (Fig. 1, TPO2 columns), with reductions of about 70% compared to pre-TPO ($\approx 3\times$ faster). Integration of these modes is ongoing and tests with larger production GUIs are still pending.

Modernization and Qt6 Support

Recent work removed deprecated dependencies and added support for newer stacks, e.g. NumPy 2.0 [13] and Qt6. Taurus 5.3 integrates PySide6 and PyQt6 while preserving Qt5 compatibility. The continuous integration (CI) matrix was set up to build and test Taurus against different combinations of Python versions and Qt bindings to prevent regressions.

EXPERIMENT ORCHESTRATION

Sardana is responsible for experiment orchestration. Below we review progress since the short and mid-term roadmap presented in [6].

Configuration Tool

The sardana-config tool now supports multiple files for improved modularity and maintainability, and provides a graph view of controllers and elements to easily visualize dependencies and detect inconsistencies (Fig. 2). To decouple development, the tool was moved to its own repository within Sardana Organization in GitLab and can be installed independently. It registers an entry point in the `sardanactl` command.

Another important pending aspect related to the configuration tool is the handling of the MacroServer environment. Specification of what to be exported is still under discussion given its heterogeneous content (from configuration to temporary and user data). Meanwhile, two additional storage backends for the environment have been added: TANGO free properties and Redis, alongside the existing `shelve` backend. This eases interaction from external clients beyond Sardana.

Continuous Scans

Multiple synchronization descriptions Each trigger/gate controller axis can now define its own synchronization description, enabling complex setups such as mixing fast detectors with passive elements (e.g. shutters) or slower devices reporting at different rates (Fig. 3).

Shutter Element A shutter object is being added to Sardana to enable automatic shutter setup within scans. It behaves as a passive element (no data acquisition), is configured via attributes and is added to the measurement group with its dedicated trigger/gate axis. A first implementation is in use at one ALBA beamline.

Data Publishing Using the BlissData library [14] developed at the ESRF, Sardana can publish experimental data to a Redis database enabling real-time data availability for external consumers (data composition, plotting, storage, analysis). By using Blissdata, clients can be shared with the ESRF enhancing collaboration between facilities. A proof of concept of its implementation in Sardana core is available and functional. This is a first step towards disabling the current recorder in favour of decoupled acquisition and reconstruction (e.g. for multiple synchronization experiments).

API Current needs on modern beamlines require complex scans. Recent custom scans at MAX IV [15] and ALBA [16] extended `GScan` and `CAcquisition` classes. While successful, they revealed code repetition and use of protected members. These findings allowed to identify the current API limitations and there are already plans to work on improving it.

Mesh Scanning The performance and reproducibility of mesh continuous scans (i.e. `meshct`, involving two motors in a snake-like trajectory) have been improved by preparing the measurement group only once at the beginning and by revising active-time calculations to correct mismatches between scan lines. Further developments include the trajectory mesh scan [17] and the `imeshct` variant, which supports scanning within custom-shaped regions [18].

Experiment Status Widget

The Experiment Status Widget, announced in the previous conference, is now available. It provides real-time status of experiment elements, helps diagnose unresponsive macros

ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-WEMG017

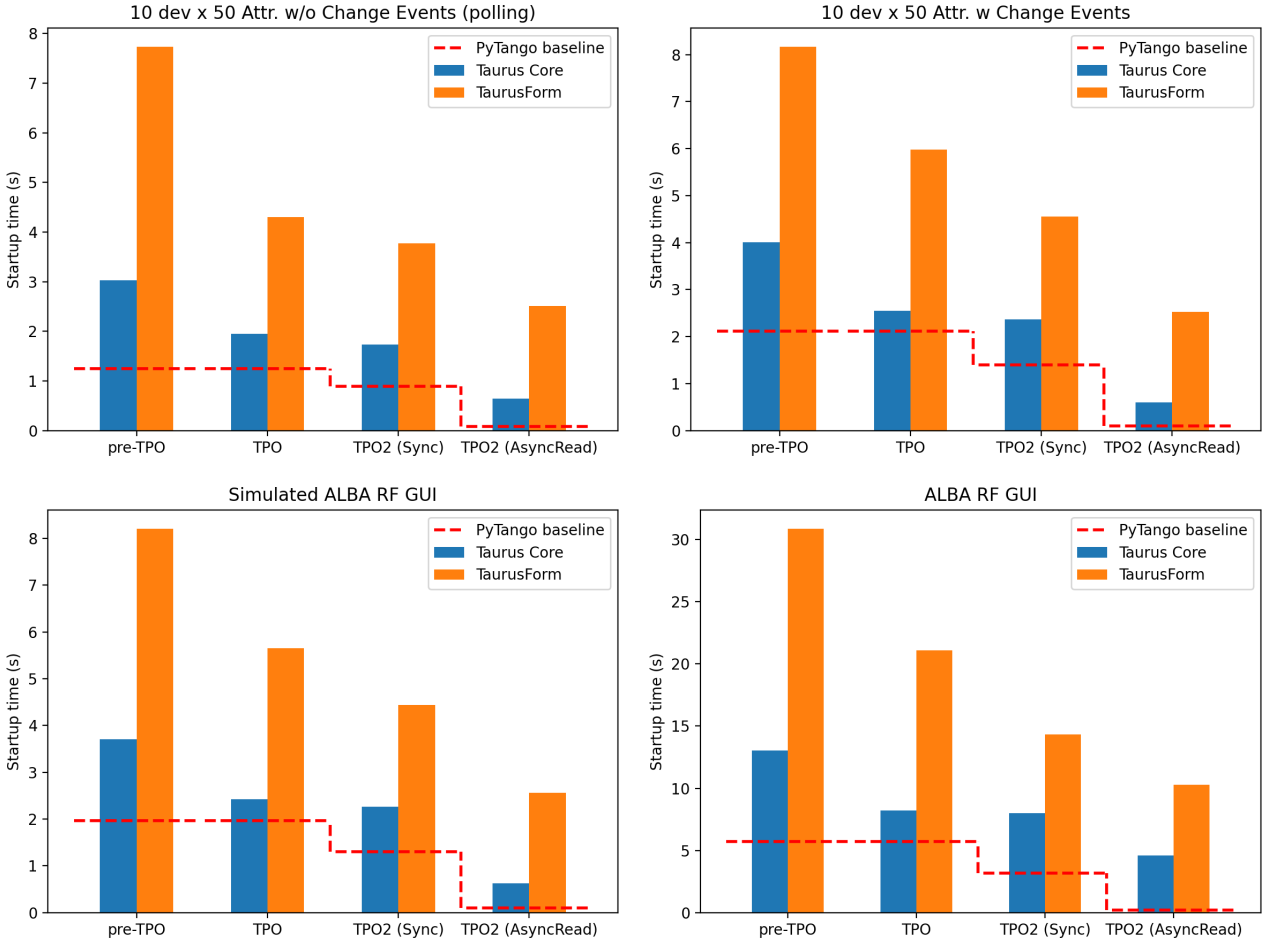


Figure 1: Startup time comparison for Taurus Core (no PyQt layer, blue bars) and TaurusForm (four Taurus Qt widgets per attribute, orange bars) before optimisation (pre-TPO) and after TPO/TPO2 (`Sync`+Taurus first read and `AsyncRead`). Top: 500 *fast* attributes (no read delays) without events (left) and with events (right). Bottom: ALBA Radio Frequency GUI with 496 attributes with events, 81 without, and 7 raising exceptions, using simulated devices (left) and real hardware (right). Red dashed lines indicate the PyTango baseline for each case (DeviceProxy creation plus subscription to configuration and change events). Benchmarks with real devices were performed on a 2-core (Xeon E5-2650) virtual machine with 4 GB RAM, while tests with simulated devices were run on an 8-core (Intel Ultra 5 135H) laptop with 16 GB RAM.

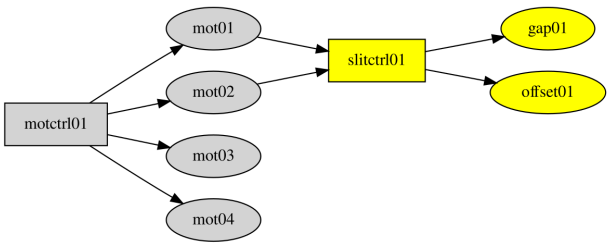


Figure 2: Graph generated by the Sardana configuration tool: relations between physical motors and a slit controller with two pseudo-motors (gap and offset).

and enables stop/abort and reconfigure actions, improving reliability and recovery times.

DEVELOPMENT PRACTICES AND SUSTAINABILITY

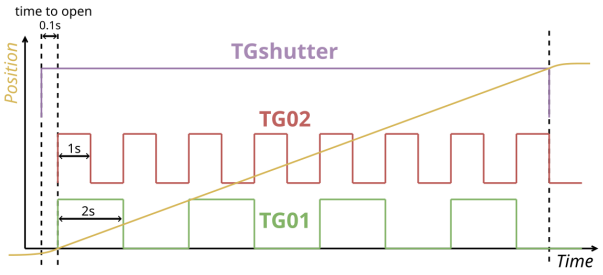
Documentation

In the last community workshops, improving the documentation has been a recurrent topic. In the case of Sardana, a recent meeting at MAX IV defined the next steps for reorganizing the documentation and the creation of more tutorials and examples, with the aim of making it more accessible to new users.

Testing, Continuous Integration and Packaging

Sardana and Taurus continue to streamline CI: lighter images, broader Python/Qt matrices and reproducible local runs. Packaging has been modularized in Sardana and Taurus (core, Qt components, tools, plugins), allowing users to install only what they need.

ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-WEMG017



Synchronization description

Synchronizer	Delay (s)	Active (s)	Total (s)	Nr Points
TG01	0,1	2	4	8
TG02	0,1	1	2	4
TGshutter	0	16,1	16,1	1

Measurement Group configuration

Channel	Synchronizer	Synchronization
twod01	TG01	gate
ct01	TG02	trigger
shutter	TGshutter	trigger

Figure 3: Multiple synchronization descriptions in Sardana: two trigger/gate elements (TG01, TG02) with different pulse trains and a passive shutter element. The top plot shows their time-domain signals together with the motor trajectory. The tables below illustrate the corresponding synchronization parameters and the measurement-group configuration linking detectors and shutter to their synchronizers.

Code Quality and Modernization

Sardana is progressively adopting type annotations to improve readability and enable early static checks. It plans also to adopt linters/formatters. Taurus already uses Black and Flake8 as part of the CI and is planning a move to Ruff.

Community and Sustainability

Beyond the technical developments, Taurus and Sardana continue to grow within the scientific community. Recent examples include the evaluation of Taurus at the ESRF and SOLEIL accelerator units, and the adoption of Sardana-Taurus at the HI-JENA laser laboratory. To support new users, the community has organized dedicated training sessions and discussions on blocking issues. In addition, both projects hold monthly open follow-up meetings as well as dedicated workshops to discuss ongoing developments and roadmaps. For instance, Sardana and Taurus workshops were organized during the TANGO meeting 2024 at SOLEIL [19], and a Sardana workshop took place at MAX IV on August 2025 [20].

These activities, together with the ongoing improvements in testing, quality checks, documentation and packaging, contribute to the long-term sustainability of both projects. They not only address immediate performance and modernization goals, but also build a cleaner, more approachable codebase that lowers maintenance costs, eases onboarding of contributors and promotes knowledge transfer across facilities.

ROADMAP

The short- and medium-term roadmaps of Sardana and Taurus are shaped by community discussions and workshops. For Sardana, and considering that some tasks are already in progress, the priorities are:

- **APIs**: Redesign of the Continuous Scan API, with lessons learned from previous custom implementations to be consolidated in a community *hackathon*. Provide a user-friendly multiple-synchronisation API.
- **Blissdata publishing**: Complete publication from the pool. Validate external saving and reconstruction with mixed-rate detectors. Disable the current recording mechanism and improve scan progress reporting.
- **Sardana Configuration Tool**: Include MacroServer environment in the file format. Define its usage workflow.
- **Shutter**: Consolidate native support within continuous scans.
- **Motor position archiving**: Push archiving events. Poll when idle.
- **Trajectories**: Evaluate native support in the core.

The mid-term roadmap includes the following features acting as a backlog:

- **APIs**: Review the Macro API and introduce type hints and make macro writing more Pythonic.
- **Experiment Setup**: Simplify channel selection and workflow of the ExpConf GUI. Add support for main-secondary relations between trigger elements.
- **Sequencer widget**: Add support for hooks in plain text, dry-run mode and retry/resume capabilities.
- **Reliability and Error Handling**: Enable user-defined recovery strategies. Provide an error visualization tool. Improve consistency (emergency brake, element states).
- **Custom commands in controllers**: Allow custom commands to be defined in controllers and exposed in Taurus widgets.
- **Laser facilities workflow**: Aadapt Sardana to better support laser experiments, which are less standardized and often require adjustments between scan points.
- **Linting and standardisation**: Adopt linters/formatters in Sardana and integrate them into CI. Introduce a merge-request checklist to enforce good practices (e.g. update changelog, documentation).
- **Documentation**: Reorganize for clarity. Provide tutorials for users. Migrate to Read the Docs [21] wit support for sub-projects (plugins and tools) and multiple versions.

ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-WEMG017

For Taurus, the developments are mainly discussed in monthly follow-up meetings. Current priorities are:

- **TPO**: Finalize TPO project with the adoption of asynchronous event subscriptions in TANGO
- **Code modernization**: Replace unmaintained `guiqwt` with `PlotPy` or `pyqtgraph`. Evaluate Wayland support. Change linter to `Ruff`.
- **Documentation**: Create a standard widgets catalog. Reorganization.
- **Issue backlog**: Address the growing queue due to more facilities involved. Consider creating bug-squashing events, as done for Sardana.

CONCLUSION

Sardana and Taurus provide an open-source SCADA solution for scientific facilities. Recent work improved performance, usability, and maintainability. The projects continue to evolve through community-driven roadmaps, with a strong focus on user demands, such as continuous scans and modern user interfaces. The projects also keep in mind the long-term sustainability, thanks to their modular architecture and strong users and developers community.

ACKNOWLEDGEMENTS

The authors thank the Sardana, Taurus and TANGO communities for continuous support, feedback, and contributions. We acknowledge the active involvement of Jose Gabadinho, Albert Olle, Steven Wohl (ALBA); Jan Kotański, Udai Singh, Yury Matveyev (DESY), Carla Takahashi, Yimeng Li, Lin Zhu, Áureo Freitas, Dmitry Egorov, Marcelo Alcocer, Anton Joubert, Hanno Perrey, Mirjam Lindberg (MAX IV); Wojciech Wantuch, Ireneusz Zadworny, Mateusz Floras, Adrianna Pytel, Michał Fałowski (SOLARIS); Wojciech Kitka, Łukasz Żytniak (S2Innovation); Reynald Bourtembourg, Natxo Vergara, Vicente Rey-Bakaikoa (ESRF); Raphaël Girardot, Patrick Madela, Alexandre Moutardier (SOLEIL); Alexander Kessler (HI-JENA) and Thomas Ives (SKAO) for their work and collaborative efforts to ensure that Sardana and Taurus remain a sustainable and relevant solution for scientific facilities.

REFERENCES

[1] Sardana, https://sardana-controls.org/

[2] T. M. Coutinho *et al.*, "Sardana: The Software for Building SCADAS in Scientific Environments", in *Proc. ICALEPCS'11*, Grenoble, France, Oct. 2011, paper WEAAUST01, pp. 607–609.

[3] Taurus, https://taurus-scada.org/.

[4] C. Pascual-Izarra et al., "Effortless Creation of Control & Data Acquisition Graphical User Interfaces with Taurus", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015, pp. 1138–1142. doi:10.18429/JACoW-ICALEPCS2015-THHC3O03

[5] Tango, https://tango-controls.org

[6] Z. Reszela et al., "Improving user experience and performance in Sardana and Taurus: A status report and roadmap", in *Proc. ICALEPCS'23*, Cape Town, South Africa, Oct. 2023, pp. 1420–1425. doi:10.18429/JACoW-ICALEPCS2023-THPDP050

[7] Qt, https://qt.io

[8] PySide, Qt for Python project, https://wiki.qt.io/Qt_for_Python

[9] PyQt, https://riverbankcomputing.com/software/pyqt/.

[10] taurus_pyqtgraph, https://gitlab.com/taurus-org/taurus_pyqtgraph/.

[11] guiqwt, https://github.com/PlotPyStack/guiqwt

[12] TANGO EventSubMode class reference, https://tango-controls.gitlab.io/cppTango/10.1.0/namespaceTango.html#adecc04f481583088538a29af2ca65c07

[13] C. R. Harris *et al.*, "Array programming with NumPy", *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. doi:10.1038/s41586-020-2649-2

[14] Blissdata, https://bliss.gitlab-pages.esrf.fr/blissdata/index.html

[15] V. Da Silva *et al.*, "Multimodal data acquisition system at MAX IV", presented at ICALEPCS'25, Chicago, IL, USA, Sep. 2025, paper THAG002, this conference.

[16] R. Homs, *et al.*, "BL31-FaXToR, hard X-ray microtomography and radiography at ALBA: current status and ongoing improvements", presented at ICALEPCS'25, Chicago, IL, USA, Sep. 2025, paper THPD039, this conference.

[17] M. Alcocer *et al.*, "Hardware orchestrated, multi-dimensional, continuous scans with the IcePAP motion controller" presented at ICALEPCS'25, Chicago, IL, USA, Sep. 2025, paper THMR007, this conference.

[18] Y. Li *et al.*, "Enhancing efficiency in high-resolution 2D mapping: arbitrary geometry scanning for μXRD/SAXS and μXRF at MAX IV", presented at ICALEPCS'25, Chicago, IL, USA, Sep. 2025, paper THBG006, this conference.

[19] 38th Tango Community meeting at SOLEIL, https://indico.tango-controls.org/event/261/

[20] Sardana Workshop at MAX IV Laboratory, https://indico.maxiv.lu.se/event/5634/

[21] ReadTheDocs, https://about.readthedocs.com/.