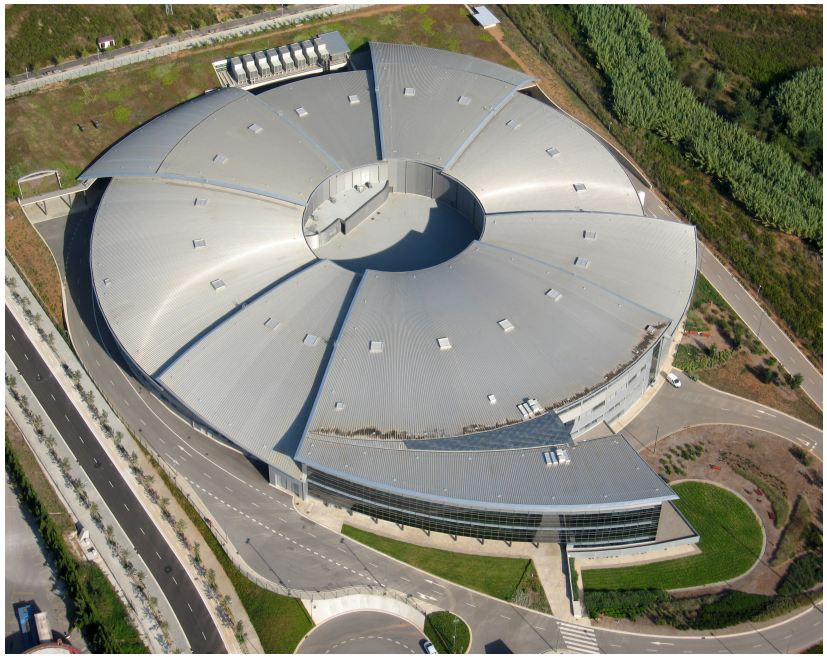
Sardana – a Python Based Software Package for Building Scientific SCADA Applications

Zbigniew Reszela

on behalf of the Alba Controls Group

PCaPAC 2014

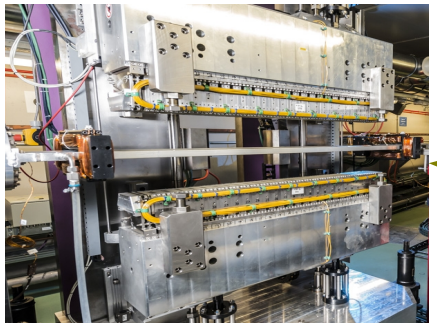


Alba Synchrotron Light Facility

Sardana logo

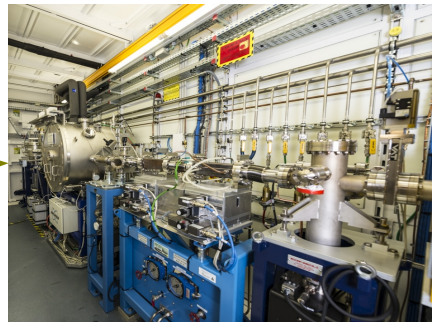
Contents

Scientific installations

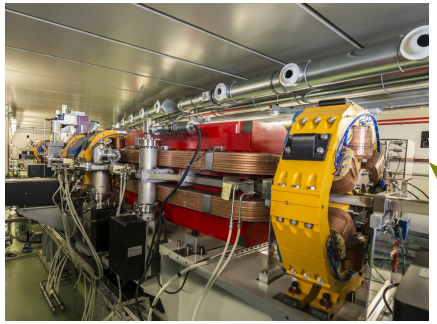


One of the ALBA insertion devices

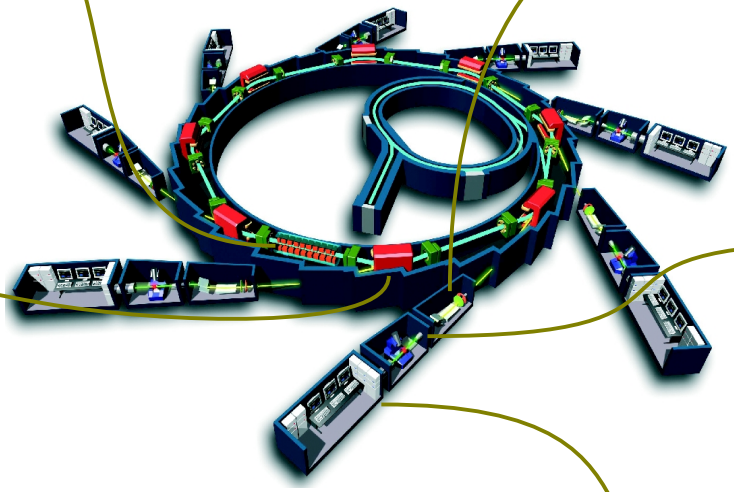
*The particle **accelerators** comprises power supplies, vacuum equipment, radio frequency stations, insertion devices and many diagnostics and actuators among others.*

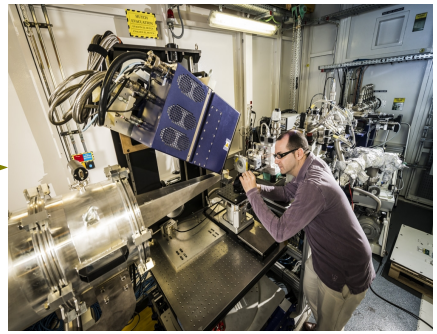


BL04 optical hutch



Magnets of the ALBA booster





BL11 experimental station

The ALBA control room

*The laboratories, e.g. a synchrotron **beamline**, usually consists of even more diverse instruments like, diffractometers, monochromators and sophisticated detectors full of moveable axes and experimental channels.*

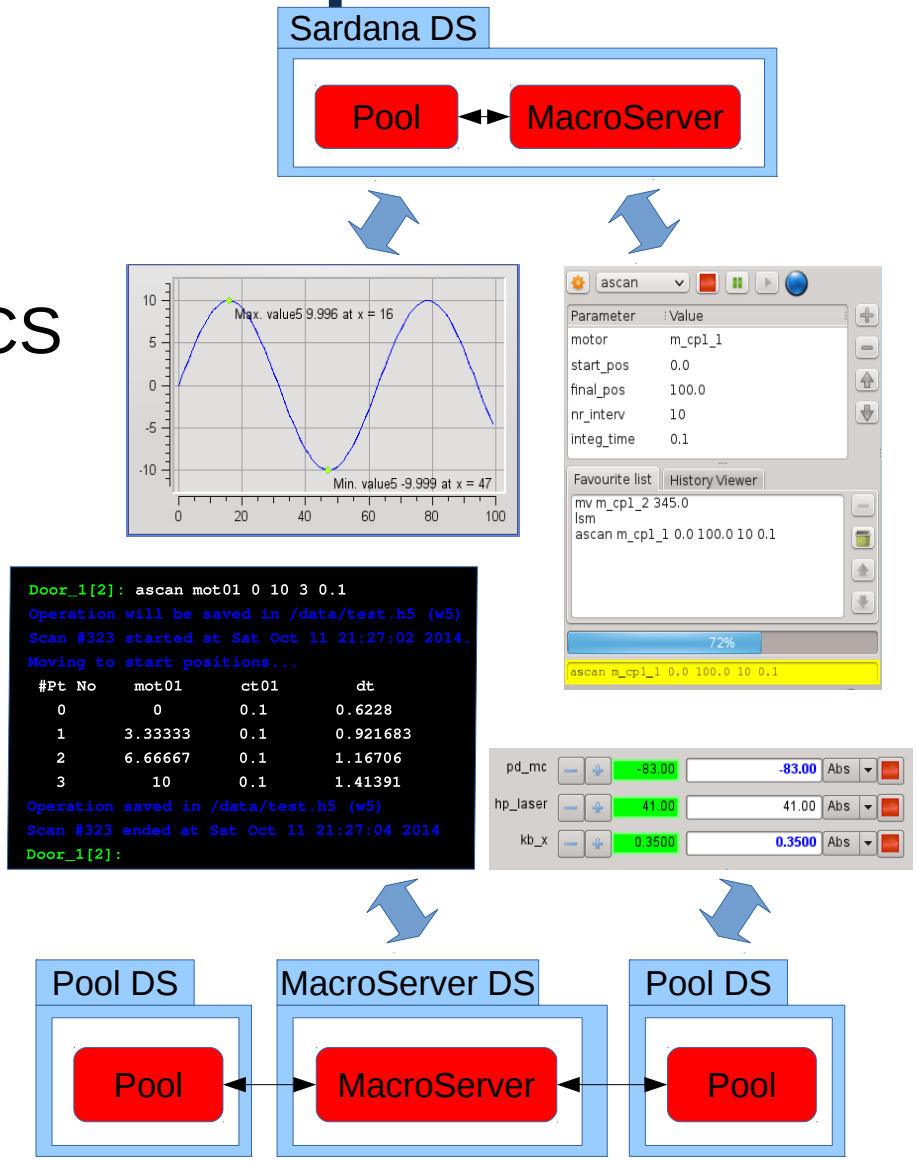
BL13 control hutch

Demands for scientific SCADA

Commercial SCADA (typically oriented to industrial applications)	**Scientific SCADA**
Does not require high flexibility. (industrial processes do not change their configuration frequently)	Beamlines require **high flexibility**. (experiment processes changes their configuration frequently)
Provide primitive sequencer.	Require **powerful sequencer** oriented to the laboratory instruments.
Favor PLC hardware solutions.	Handles **heterogeneous hardware** using different communication protocols and of different vendors.
Data acquisition is used mostly for monitoring and is stored in DB (data-timestamp pairs)	Performs **synchronized data acquisition** processes and stores data in the **specific file formats**.
Most control actions are performed automatically by RTUs or PLC.	Most control actions are performed on **operators/scientists request**.
	Requires **specific capabilities** e.g. diffractometer control.

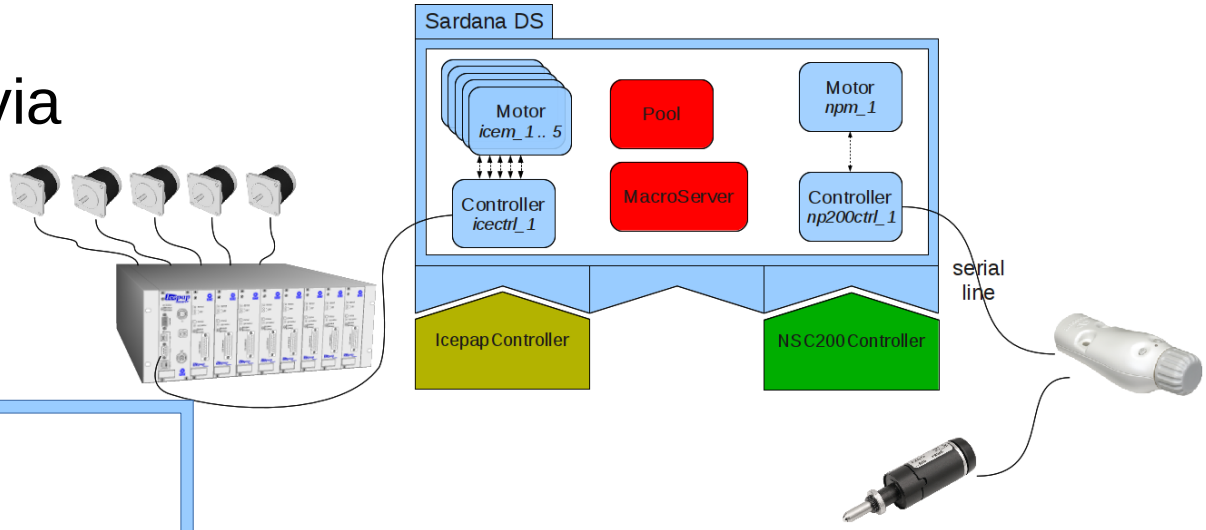
Sardana and its components

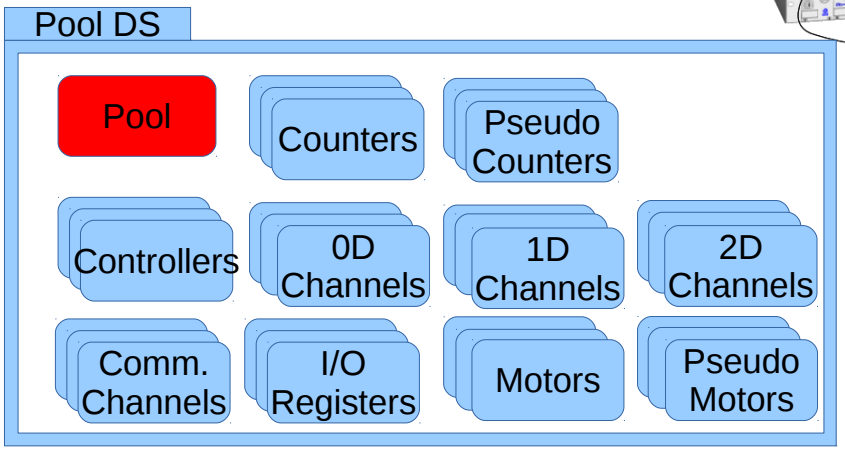
- SCADA software suite
- It is open source (LGPL)
- Allows building distributed CS
- Uses Client - Server model
- Communicates via Tango
- Server:
  Device Pool (HW access)
  MacroServer (sequencer)
  Sardana (all in one)
- Client: Taurus HMIs



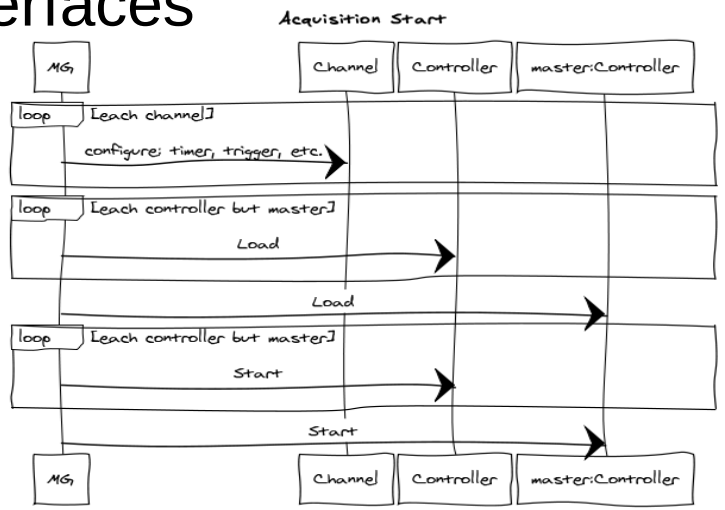
Device Pool

- Hardware Access via Controllers



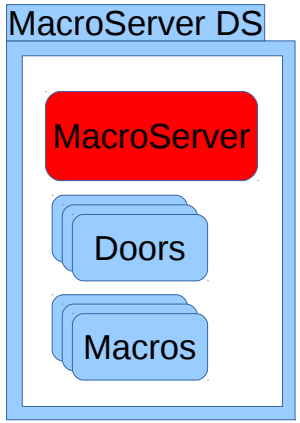


- Common & Dynamic Interfaces

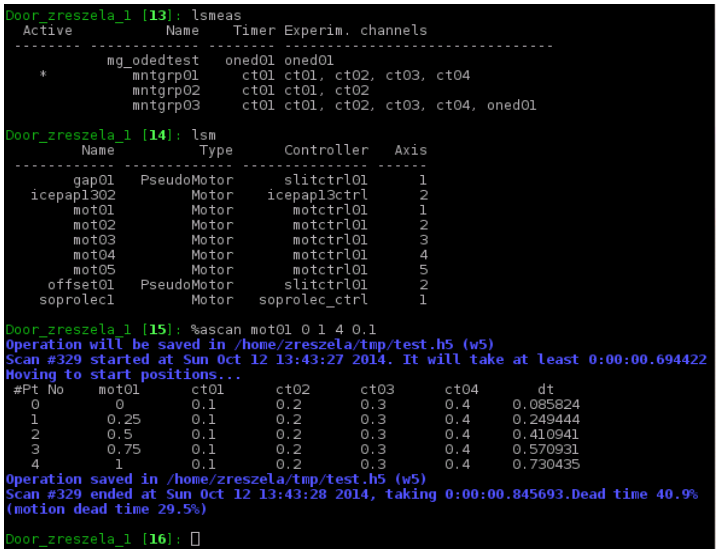


- Control algorithms: Motion, Acquisition

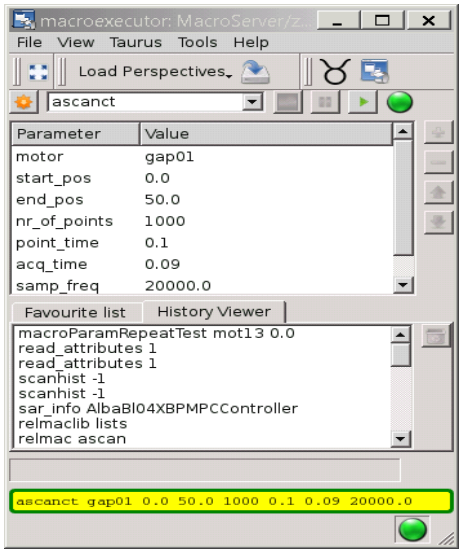
MacroServer



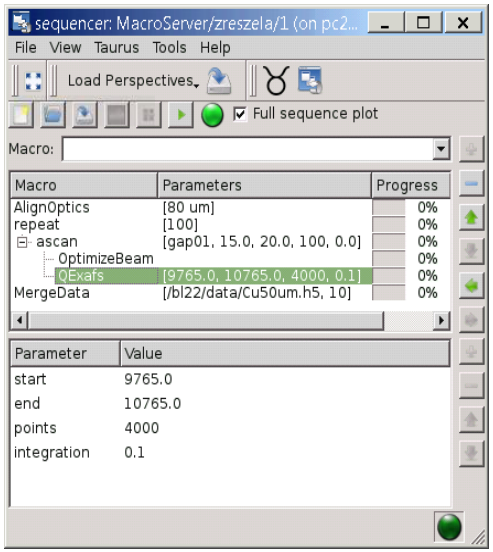
- Sequencer capable to plan and execute user procedures - ***macros***
- MacroServer & Door(s) (via Door different client applications access the MacroServer engine)
- Allows sequential & simultaneous execution of macros
- Various GUI and CLI clients can control the macro execution



Spock (based on IPython)



MacroExecutor



Sequencer

Macros

Hooks

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

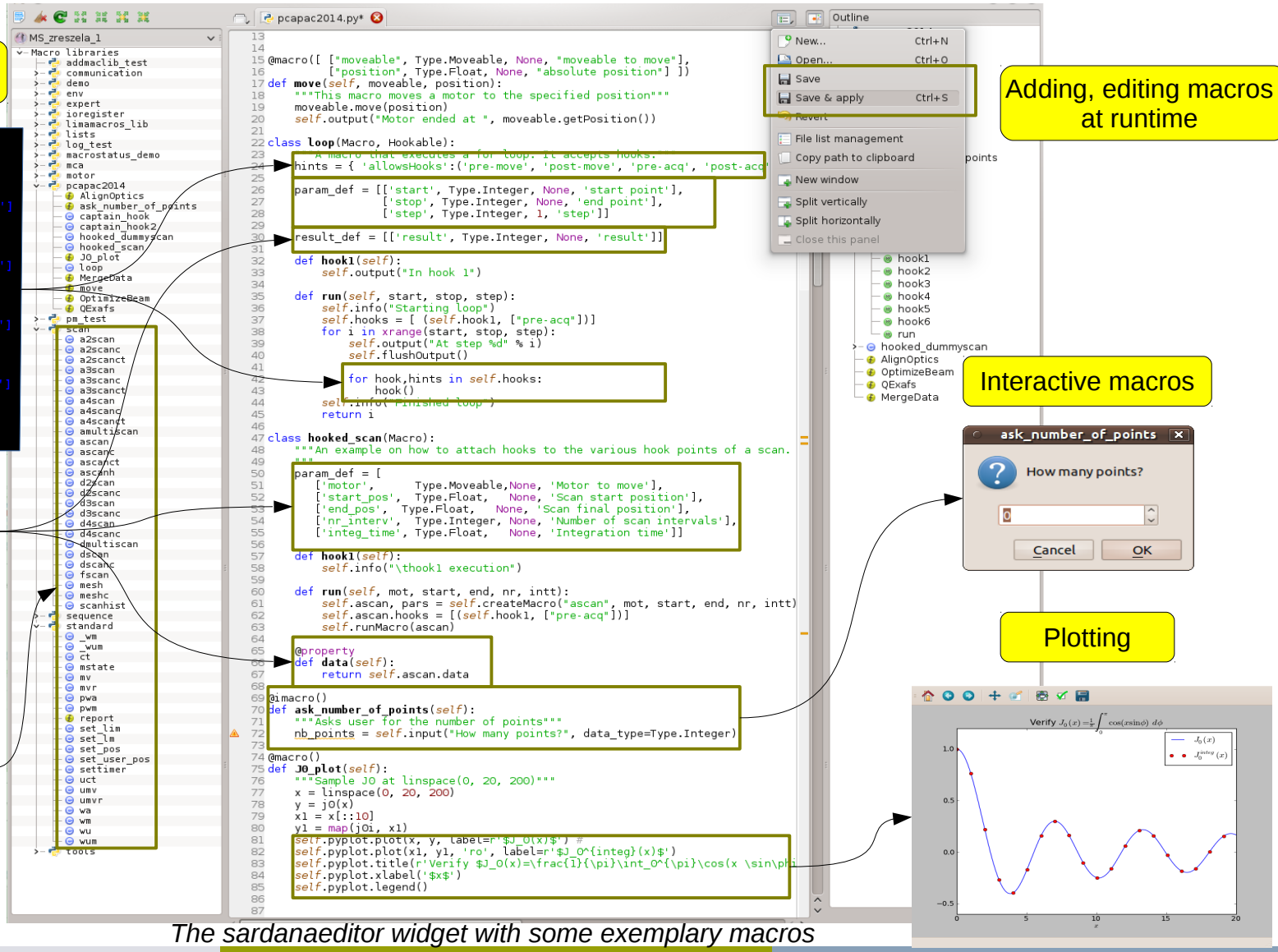
Input parameters & results & data

SPEC like commands

Adding, editing macros at runtime

Interactive macros

Plotting

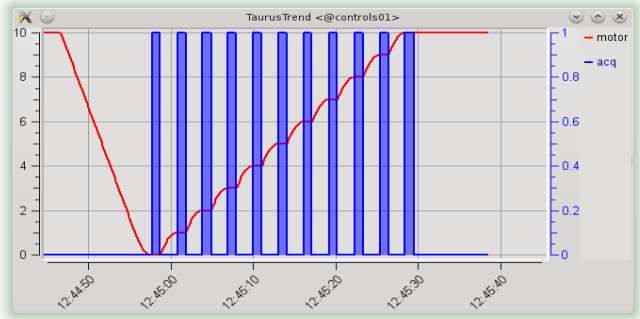


The sardanaeditor widget with some exemplary macros

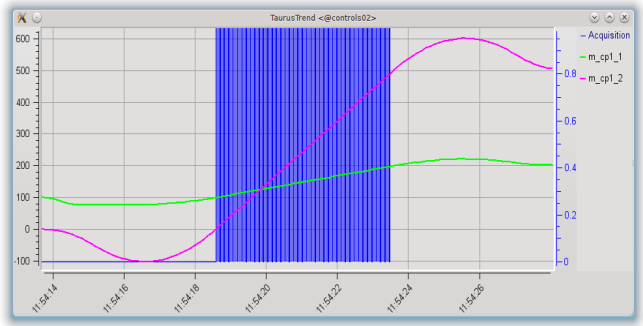
Scans & Data Handling

*The scan process consist of **data acquisition** of the involved experimental channels while **varying the scanning variable** (typically a moveable object).*

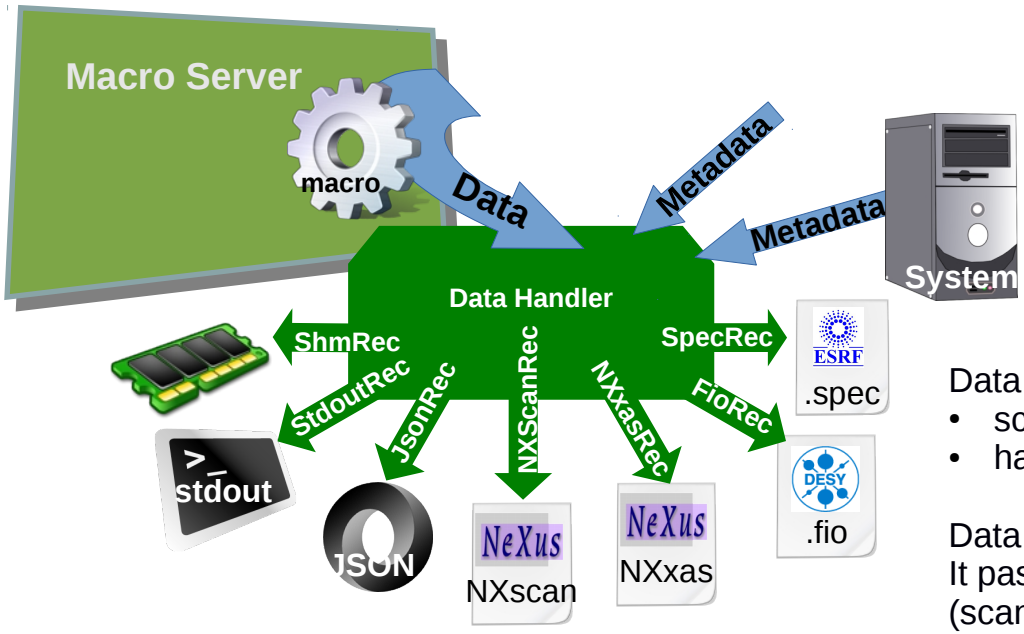
- Sardana offers various scanning modes: step, hybrid, continuous.
- Turnkey n-dimensional scan macros exists.
- Motion, data acquisition & storage are synchronized and optimized.



Motion & acquisition during the step scan.



Motion & acquisition during the continuous scan.

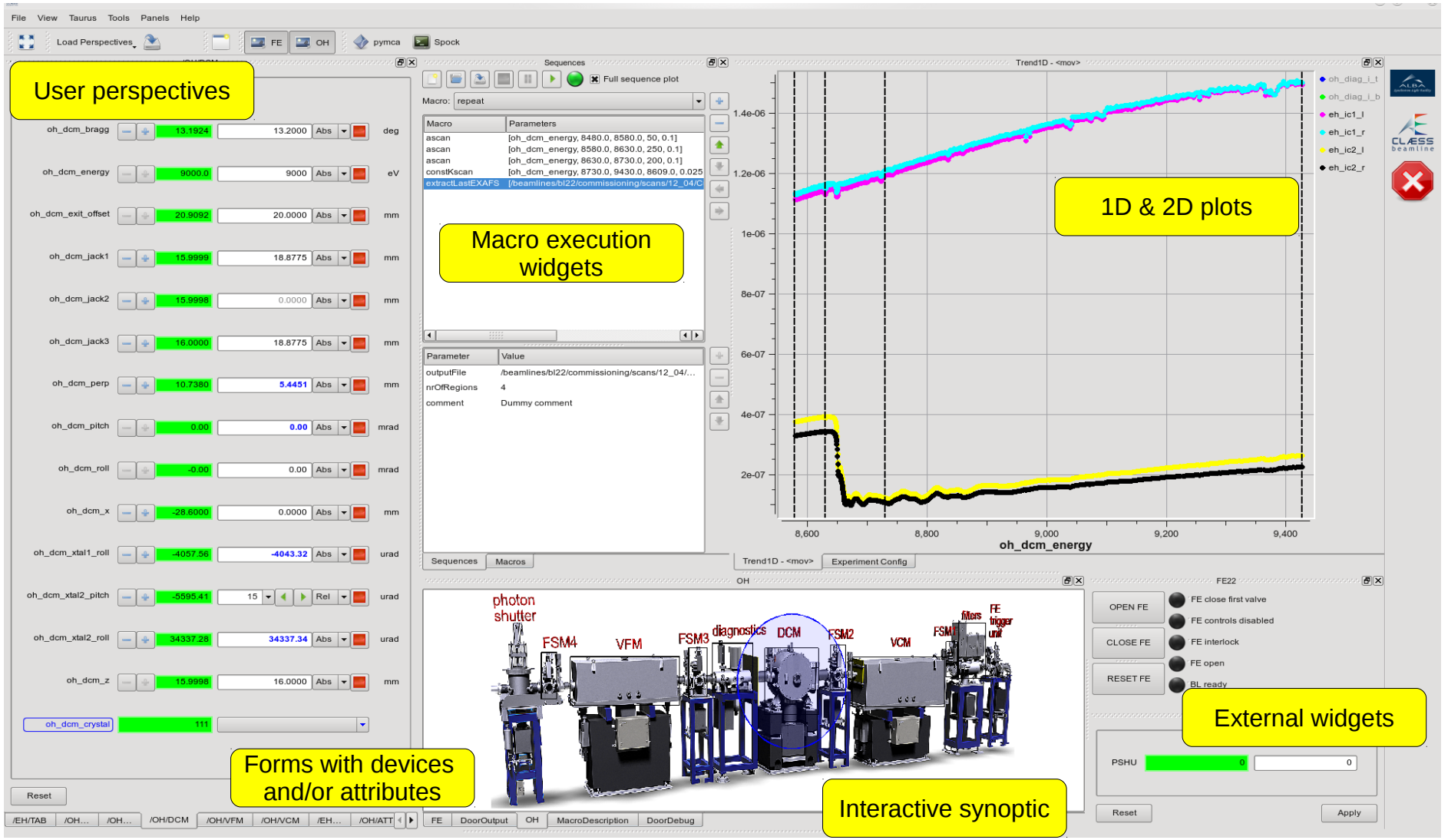


Data handing in Sardana:

- scan data can be stored in **different formats**
- handled by one or more **optional *recorders***

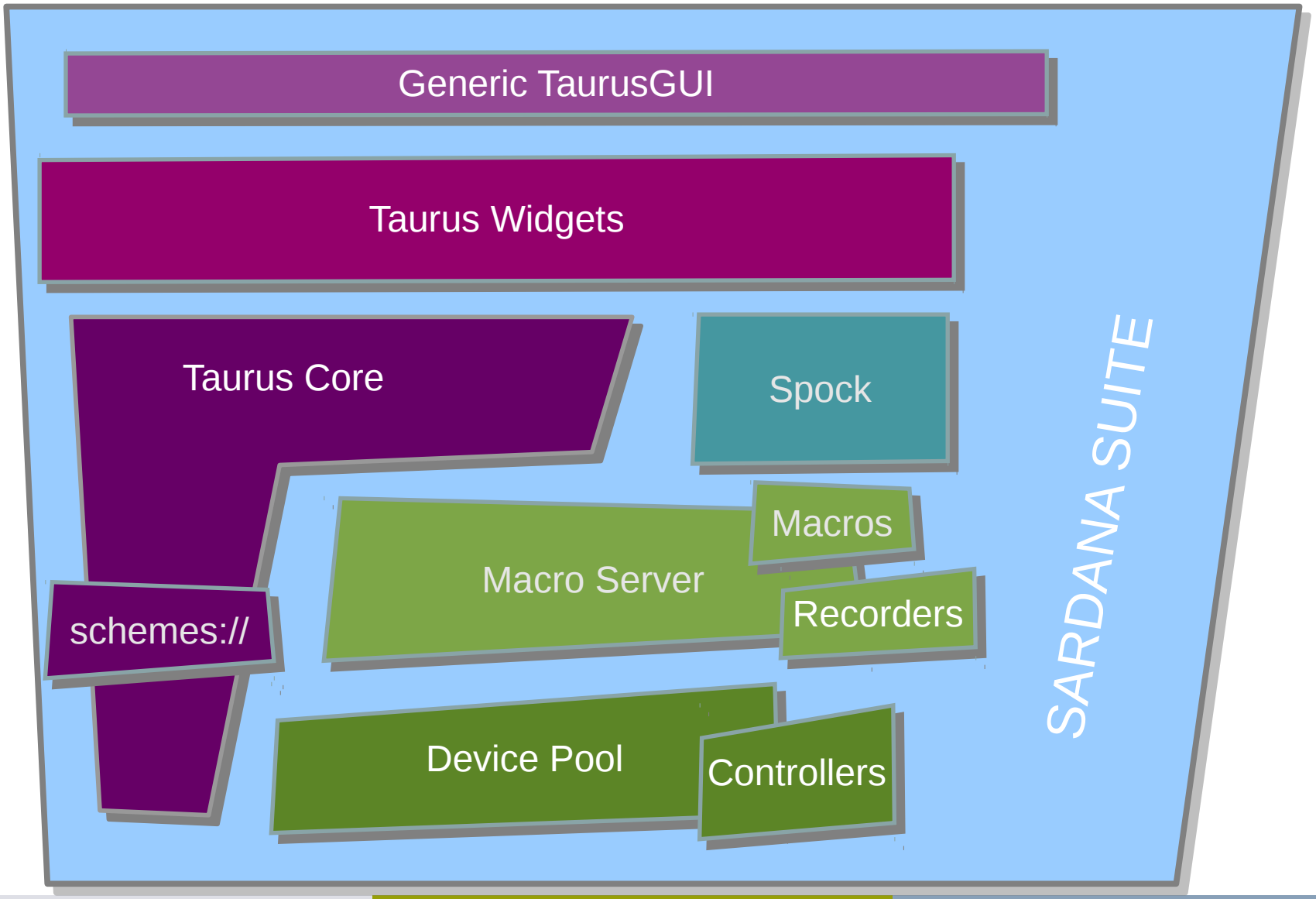
Data handler receives and gathers *metadata*.
It passes data and the experiment metadata from the (scanning) macro and despatch them to the destination in the correct format: file (nexus, spec, etc.), console output or data post-processing program, etc.

Taurus based HMIs

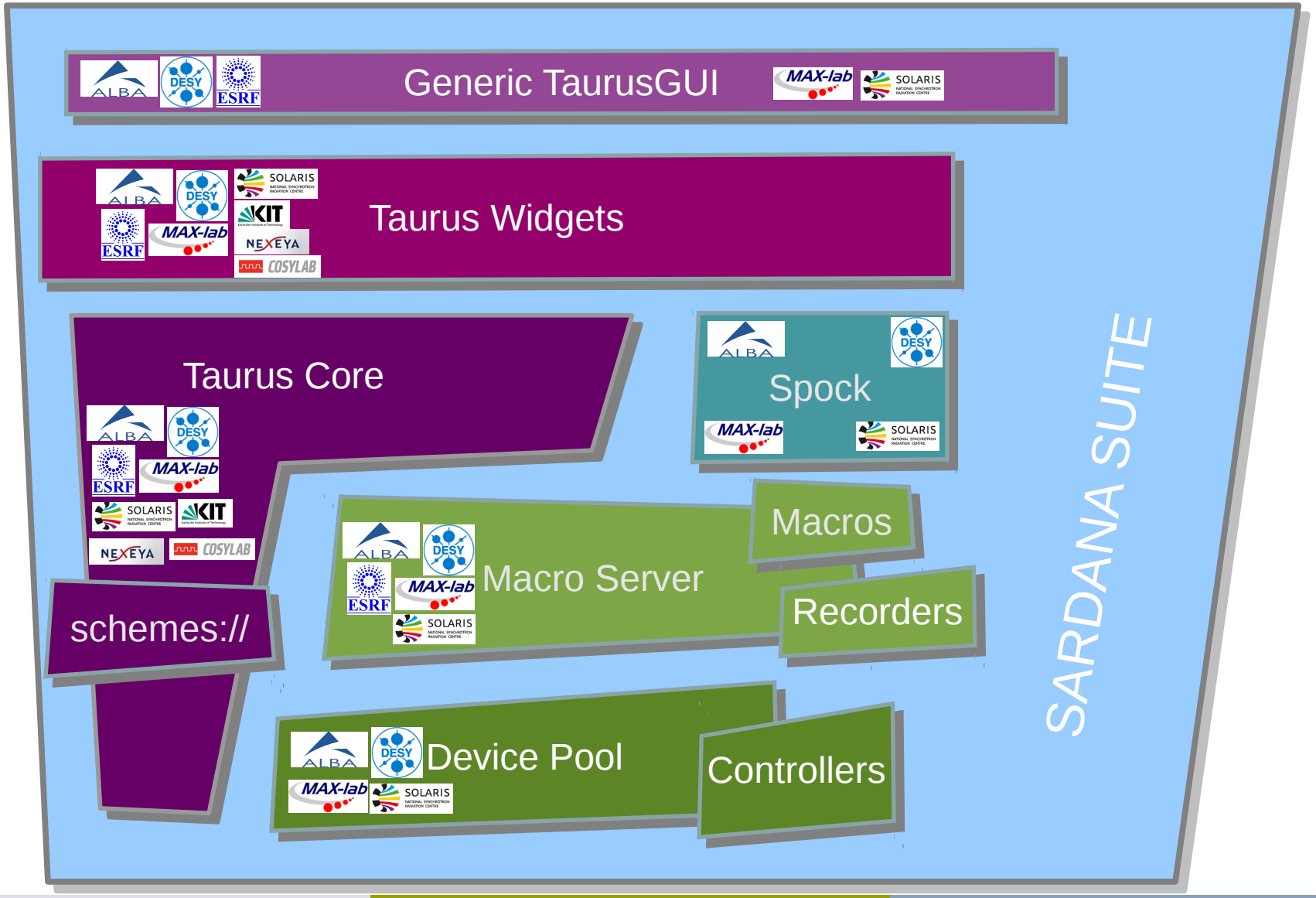


Beamline 22 GUI developed with Generic TaurusGUI Framework

Sardana Suite



Sardana Community



Current and future developments

General

Plug-in system

Use ReadThe Docs for documentation

Create the h5file:// scheme

Support Python3

Taurus

Taurus core Tango independent

Use of Pint Quantities

Multi-models (adapter pattern)

Replace Qwt for plots

Allow external logging (SEP8)

Use of standard Enum (SEP12)

Direct registering of Icons (avoid resource files)

Merge TaurusConfiguration into TaurusAttribute

Support PySide and Qt5

Generic support for archiving values

SVG synoptic view

Sardana

Generic continuous scans (SEP6)

Trigger/Gate elements (SEP6)

Controllers with heterogeneous elements

Diffractometer control (SEP4)

Pseudo roles on instance level

2D detectors: Lima integration (SEP2)

Configuration tool

Easier debugging

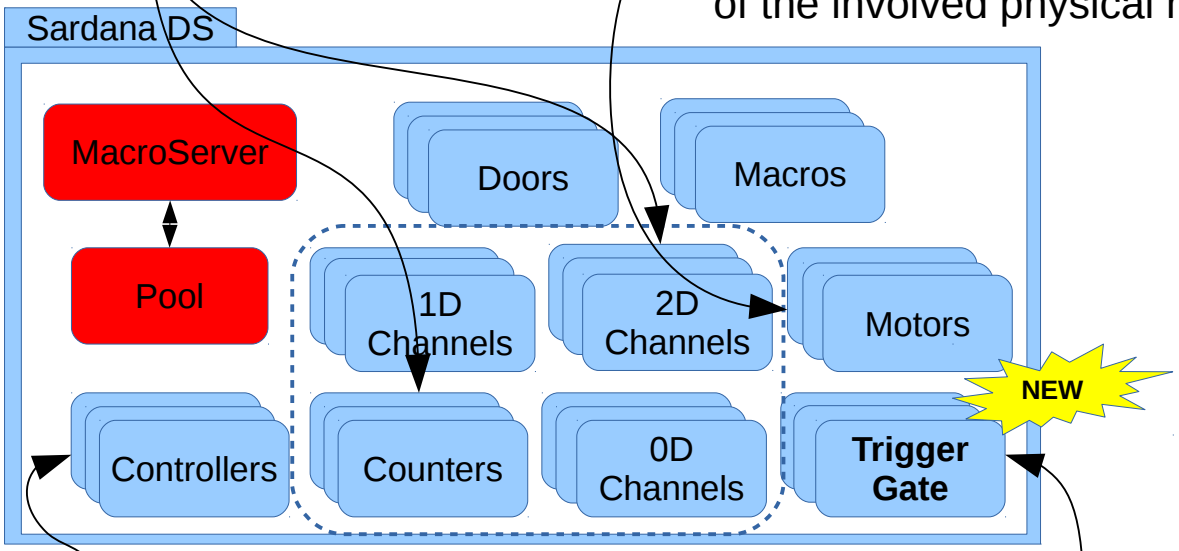
Acquisition with multiple Pools

Refactor experiment configuration

Generic Continuous Scans

Transparency between different synchronization modes of the experimental channels must be achieved. (timestamp used for triggering and data merging)

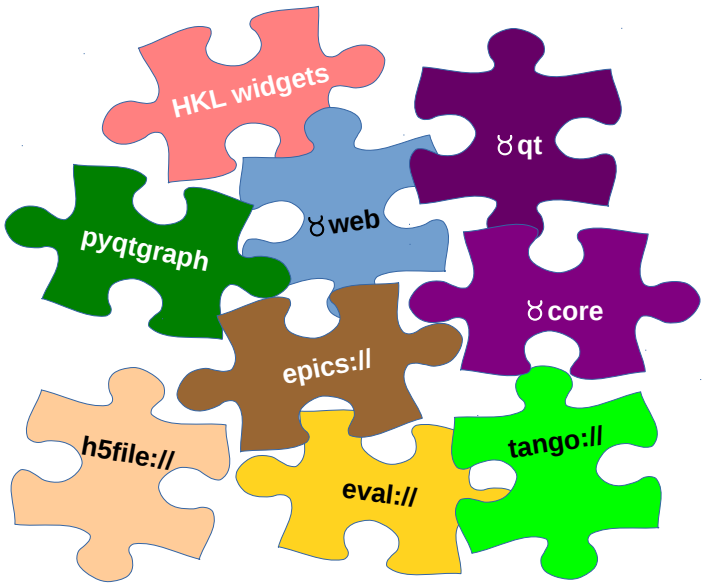
To ensure scans of the pseudomotors with non-linear formulas, with a constant rate, control of the motion trajectories of the involved physical motors is required.

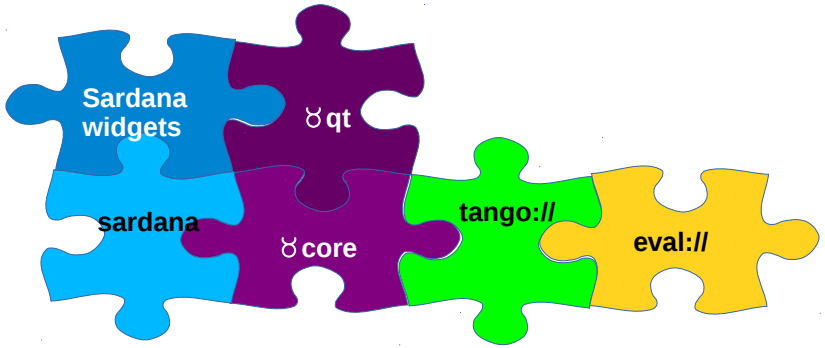


The Controller concept needs to be extended to allow handling elements of different types (multipurpose hardware)

A new element type – trigger/gate is necessary. Its interfaces must allow configurations of equidistant or arbitrary sequences of events in time or distance domains.

Generic plugin system





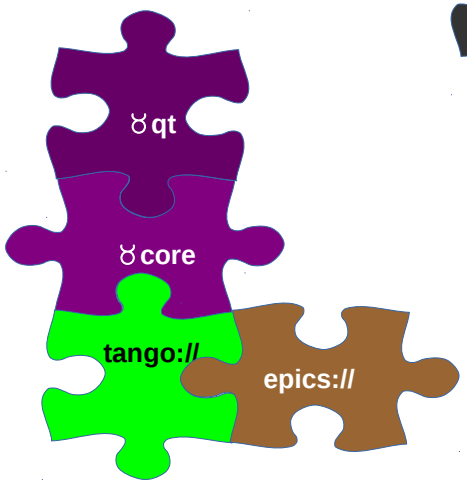
***Example**: Taurus+Sardana as we use it now in ALBA*

Plugins will make Sardana & Taurus...

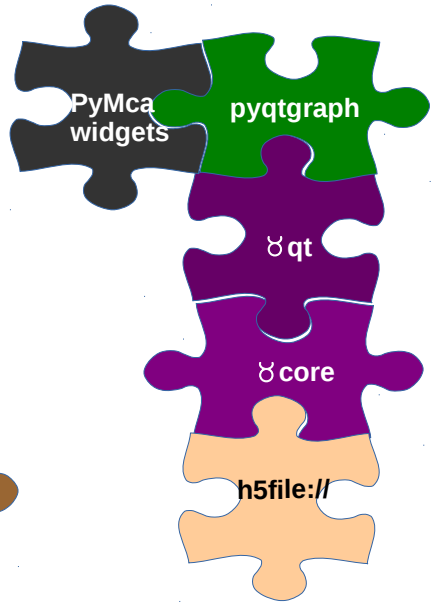
- Light: most dependencies optional
- Extendable for user specific need
- Taurus usable as a library for data analysis GUIs (not connected to control system at all)

What could be a plugin in Sardana & Taurus:

- Schems, Widgets, Taurus core extensions...
- Macros, Controllers, Recorders...



***Example**: Controlling a mixed Tango+EPICS environment*



***Example**:Taurus for Data Analysis (no control system)*

Contributors

- Alba Controls Group
- Alba scientists
- Contributors & Experts from:
  - Desy
  - ESRF
  - MaxLab
  - Solaris
  - Soleil