

A Light for Science

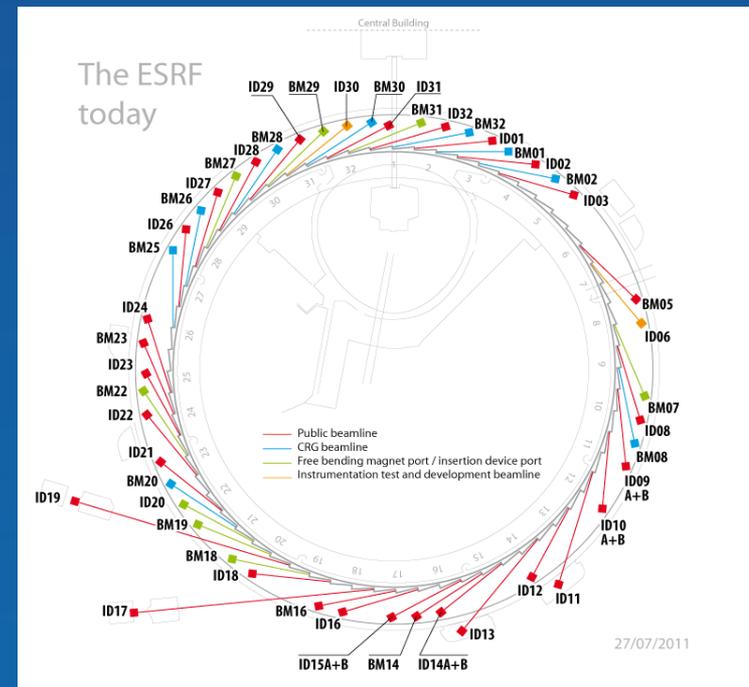


European Synchrotron Radiation Facility

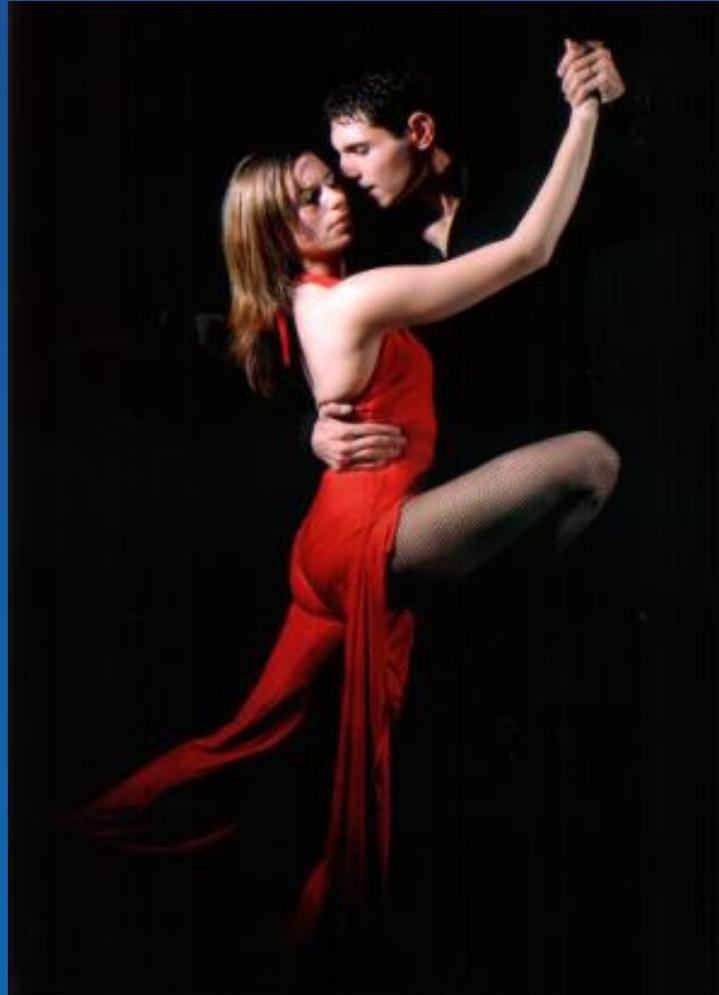


# Tango for Experiment Control

- What is Tango
- Scanning and Sequencing
- Diffractometers
- 2D Detectors
- Data Analysis Workbench



## What is Tango

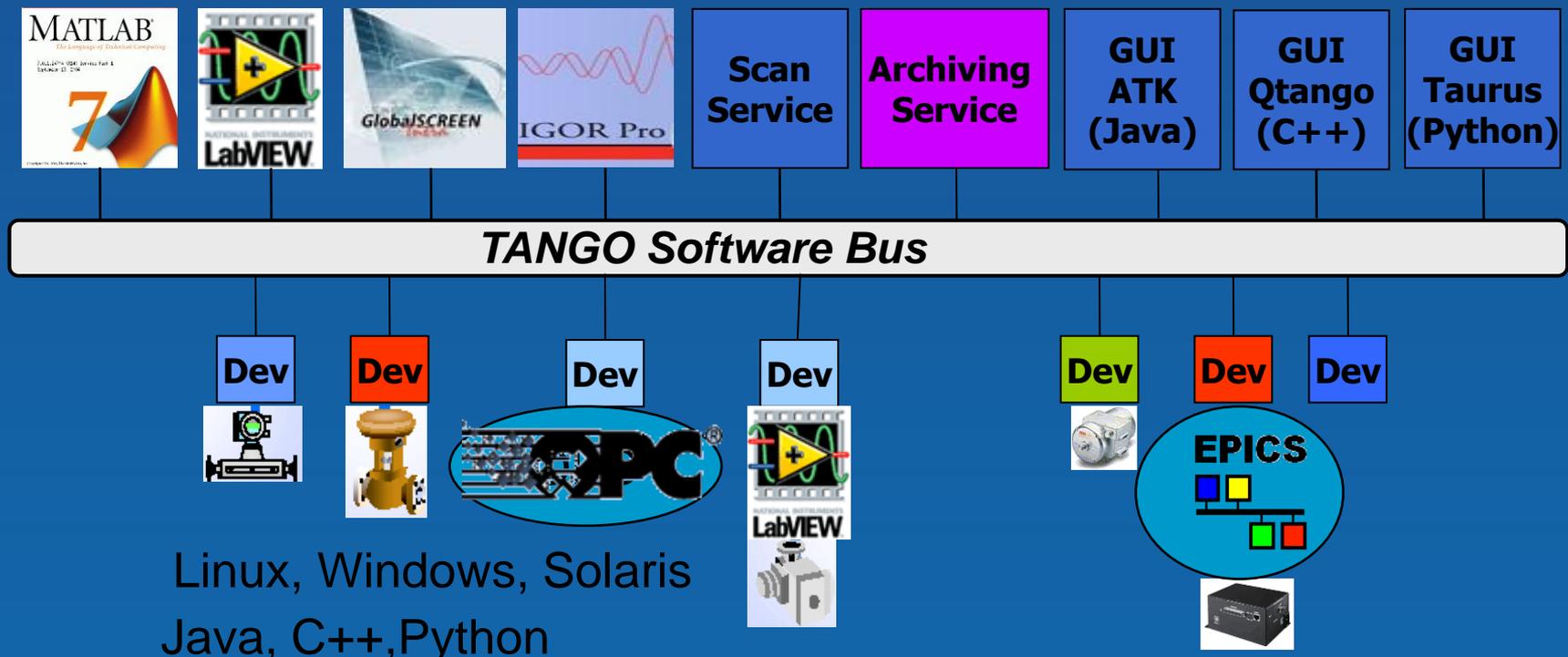


## What is Tango

- A software bus for distributed objects

Linux, Windows, Solaris

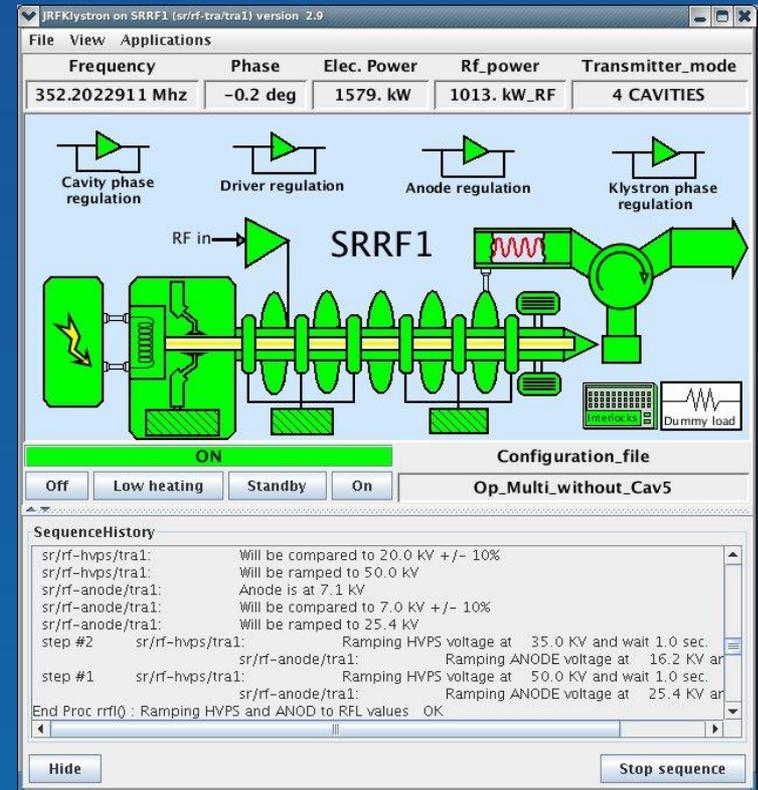
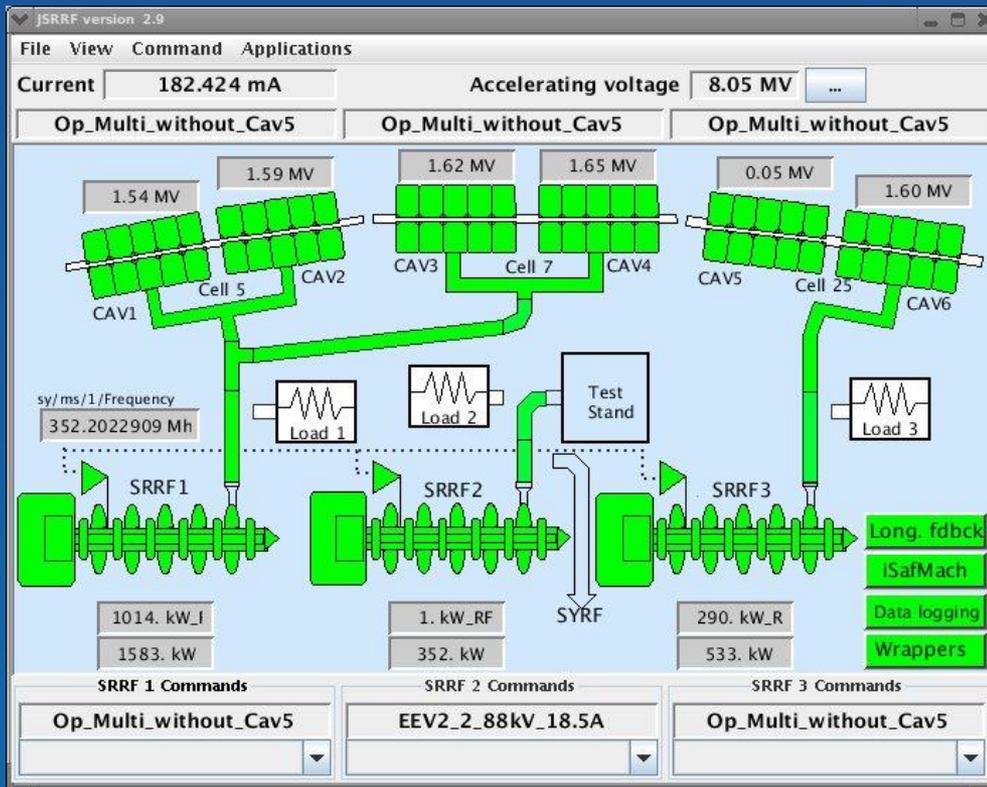
Java, C++,Python



Linux, Windows, Solaris  
Java, C++,Python

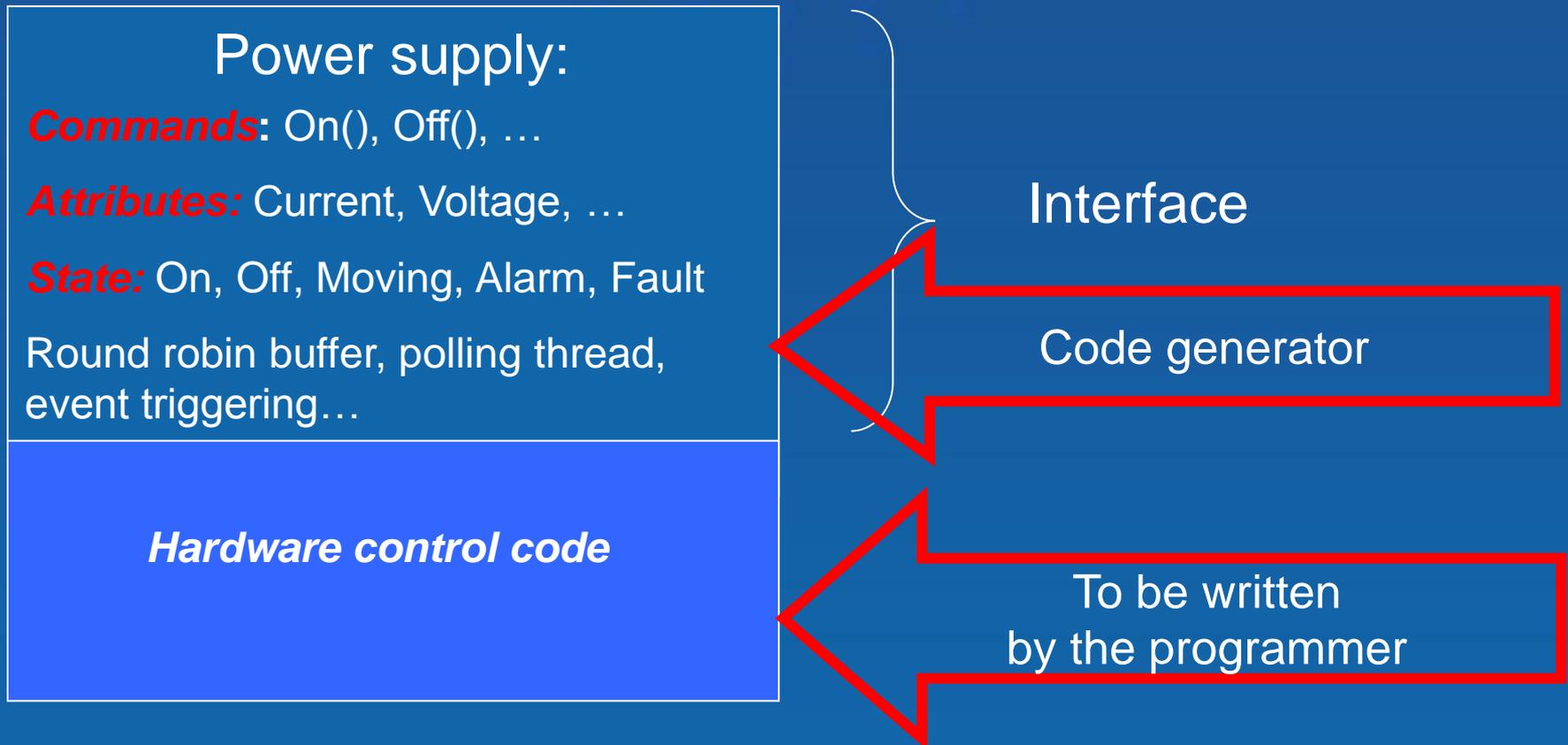
## What is Tango

- GUI frameworks for C++, Python and Java
- Synoptic editor



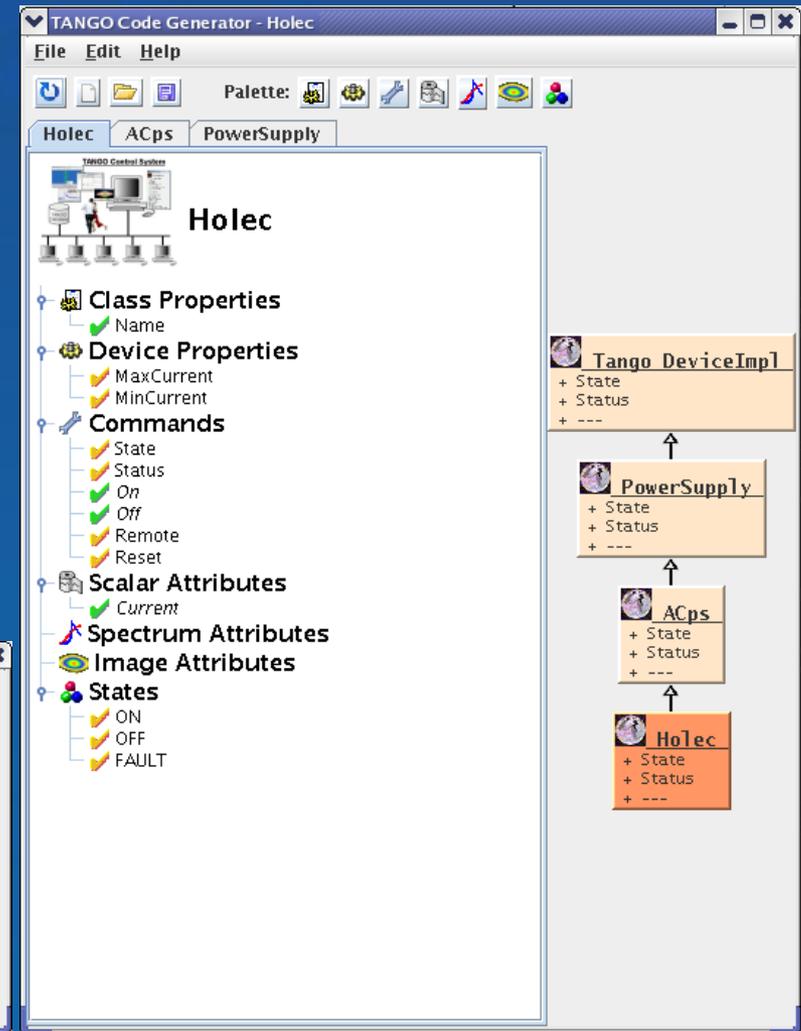
## What is Tango

### A simple Tango device



## What is Tango

- Graphical interface and state machine design
- Code generation: C++, Java and Python
- Editing and code re-generation
- Fast development cycle



## What is Tango

- Administration and survey system
- Graphical system configuration



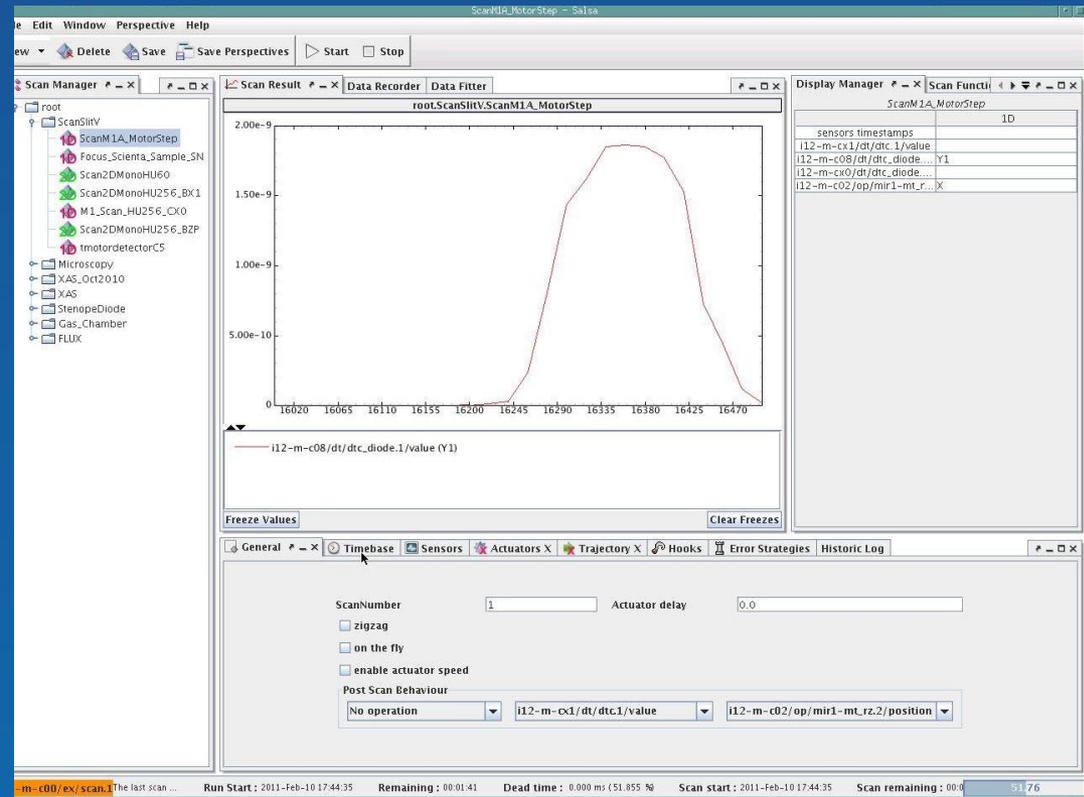
## What is Tango?

- An object oriented software bus
  - Communication types: Synchronous, asynchronous, grouped asynchronous and event driven
  - Servers and clients can be written in C++, Python and Java
- The Tango tool chain: Software from the hardware interface to the GUI

Module	Description
Core Libraries	Client/Server communication libraries for C++, Python and Java
Device Classes	About 300 hardware interface classes are available to download
GUI Frameworks	Available for C++ and Python using QT, for Java using Swing and a web interface written in PHP
Client Bindings	LabView, Matlab and IgorPro
Tools	Pogo – Code generator for device classes in C++, Python and Java Jive – Configuration and testing tool Astor – Administration and survey of the Control system
Archiving	Archiving and snapshot system with GUIs and web interface. Usable with Oracle and MySQL
Sardana	Framework for experiment control : Interface standardization, configuration, sequencing, command line interface

## Sequencing and Scanning

- Tango Scan Server + GUI
  - Different types of scans available
  - Plug-in architecture for hardware integration
  - Developed at SOLEIL

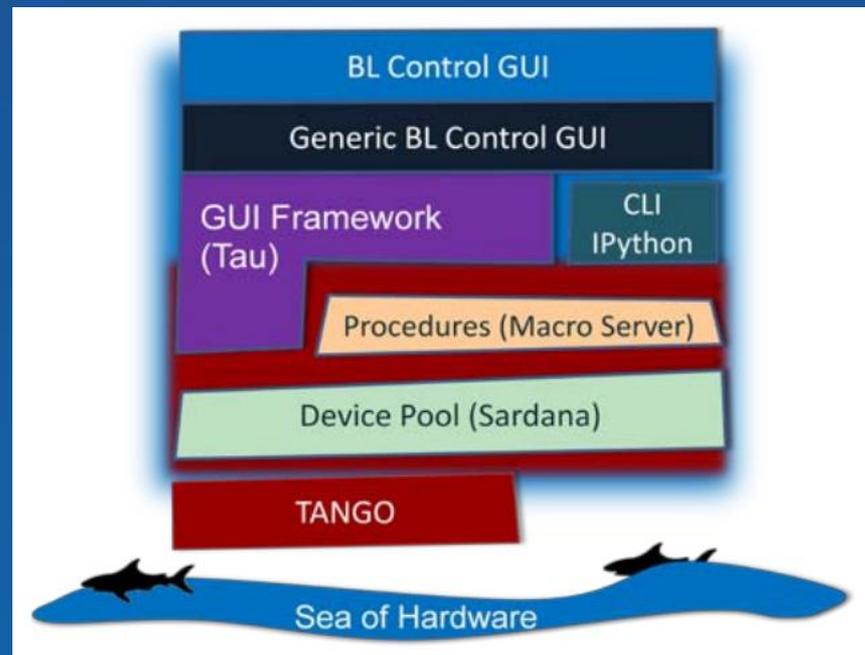




# Sequencing and Scanning

- Sardana

- Integrated user environment developed at ALBA
- Interface standardization
- Scanning framework
- Python procedures + IDE
- Command line interface
  - SPEC like syntax
- Configuration GUI
- GUI builder
  - based on PyQt widgets
- <http://www.tango-controls.org/static/sardana/latest/doc/html/index.html>



## Sequencing and Scanning

- Sardana configuration GUI



The screenshot displays the Sardana configuration GUI. The main window shows a tree view on the left with a hierarchy: controls01:10000 > Tiagos Pool > Controllers > tc\_uxtimerctrl1, tc\_simumotctrl1, tc\_simucotctrl1, diffrac4CCtrl, and IcePAPCtrl1. The main area shows the configuration for the 'Tiagos Pool' component, including tabs for Measurement Groups, Communication Channels, Controller Classes, and Experiment Channels. A 'Create Controller' dialog box is open in the foreground, showing the configuration for a 'PseudoMotor' controller. The dialog includes a description, a 'Details' section with a photo of a diffractometer, and a 'Properties' section with various parameters.

**Component:** Tiagos Pool  
tcoutinho/pool/01

**Controllers:** tc\_uxtimerctrl1, tc\_simumotctrl1, tc\_simucotctrl1, diffrac4CCtrl, IcePAPCtrl1

**Motors:** tth, th, chi, phi, H, K, L

**Motor Gro.:** tcoutinho/φ., tcoutinho/φ., tcoutinho/φ., tcoutinho/φ.

**Experiment Channels:** tcoutinho/φ., tcoutinho/φ., tcoutinho/φ., tcoutinho/φ.

**Create Controller Dialog:**

**Description:** This is the C++ pseudo motor controller for a four circle vertical diffractometer using the hkl library des

**Name:** [Empty]  
**Type:** PseudoMotor  
**Library:** Diffractometer.la  
**Class:** Diffrac4C

**Properties - Pseudo Motor Roles:**

direction:	1 0 0
gamma:	90
a:	2.84
wavelength:	2.84
c:	2.84
OperationMode:	Bissector
beta:	90
b:	2.84

**reflections:** [Empty]

**Details:** Organization: CELLS - ALBA, Family: Diffractometer, Model: Four Circle

**Buttons:** Refresh, Create, Close

## Sequencing and Scanning

- Sardana procedure IDE and the command line interface

The image displays two software interfaces used for experiment control: MacroGUI and IPy Notebook.

**MacroGUI (Left):** Shows a 'Trend1D' plot for motor 'mot01' with a scan sequence. The plot shows a step function where the motor position is constant at 0.0, 0.1, 0.2, and 0.3 for intervals of 10 units. Below the plot is the 'SardanaEditor' with a Python script for an 'ascan' macro:

```

class ascan(aNscan, Macro):
    """Do an absolute scan of the specified motor.
    ascan scans one motor, as specified by motor. The motor start
    position given by start_pos and ends at the position given
    by final_pos. The step size is (start_pos-final_pos)/nr_interv. The number
    of scans will be nr_interv+1. Count time is given by time which if
    positive specifies seconds and if negative, specifies monitor counts.
    """
    param_def = [
        ['motor', Type.Moveable, None, 'Moveable to move'],
        ['start_pos', Type.Float, None, 'Scan start position'],
        ['final_pos', Type.Float, None, 'Scan final position'],
        ['nr_interv', Type.Integer, None, 'Number of scan intervals'],
        ['integ_time', Type.Float, None, 'Integration time']
    ]
    def prepare(self, motor, start_pos, final_pos, nr_interv,
                **opts):
        self.prepare([motor], [start_pos], [final_pos], nr_interv,
                    **opts)
    
```

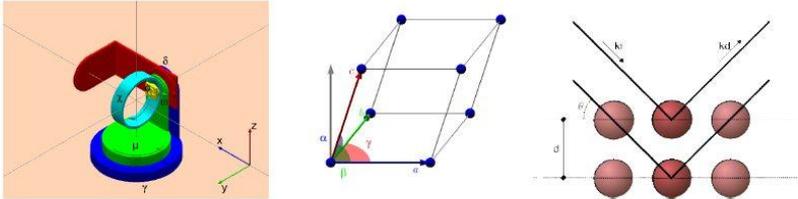
**IPy Notebook (Right):** Shows the execution of the 'ascan' macro. The command is `ascan mot01 0 100 10 0.1`. The output shows a table of scan data:

#Pt	mot01	ct01	ct02	ct03
0	0	0.1	0.2	0.3
1	10	0.1	0.2	0.3
2	20	0.1	0.2	0.3
3	30	0.1	0.2	0.3
4	40	0.1	0.2	0.3
5	50	0.1	0.2	0.3
6	60	0.1	0.2	0.3
7	70	0.1	0.2	0.3
8	80	0.1	0.2	0.3
9	90	0.1	0.2	0.3
10	100	0.1	0.2	0.3

The notebook also shows the operation saved in `/tmp/hh.h5 (w5)` and the scan #25 ended at Wed Apr 25 14:12:41 2012, taking 0:00:16.873967 (dead time was 93.5%).

## Diffractometers

- SOLEIL developed a C-library for reciprocal space transformations.
- The purpose of the library is to factorise single crystal diffraction angles computation for different kinds of diffractometer geometries.
- <http://people.debian.org/picca/hkl>

THE equation

$$RUB\vec{h} = \vec{Q} = \vec{k}_f - \vec{k}_i$$

# Diffractionmeters

- The main features are:
  - Mode computation
  - UB matrix computation
  - Crystal lattice refinement
  - Pseudo axes ( $\psi$ , eulerians,  $q$ , ...)
- Today the HKL library can handle 5 different geometries:
  - 2 circles
  - Eulerian 4 circles
  - Eulerian 6 circles
  - Kappa 4 circles
  - Kappa 6 circles
- Modes hkl
  - bissecteur
  - constant  $\omega$ ,  $\chi$ ,  $\phi$

## Diffractometers

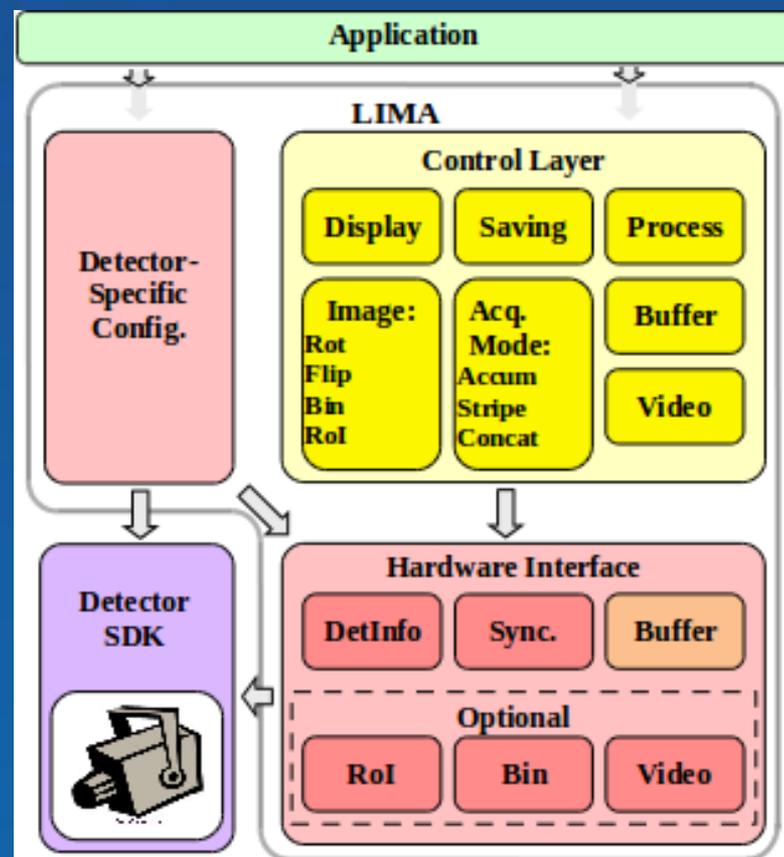
- Generic diffractometer device server + GUI
- Example : Kappa 6 Circles diffractometer GUI

The screenshot shows the 'LPS Control Acquisition Application' window. The main interface is divided into several functional areas:

- Reciprocal Space:** Displays H, K, and L values, each with a green '0.000 N' field and a 'set(H,K,L)' button.
- Axes:** Shows parameters for  $\mu$ ,  $k$ ,  $\delta$ ,  $\omega$ ,  $\phi$ , and  $\gamma$  with numerical input fields and units.
- Source Configuration:** Includes a 'Mode' dropdown set to 'simulated' and a 'Sim...' button.
- Calcul Mode:** Set to 'Vertical Eulerian 4C Bisector' with a note: 'Omega = 2theta / 2. there is no parameters for this mode.'
- Crystal Configuration:** Shows 'Crystal' as 'CuGeO3' and a list of 'Lattices' (A, B, C, Alpha, Beta, Gamma) with checkboxes.
- Reflections:** A table with columns for 'Index', 'H', 'K', 'L', 'relevance', 'affinem.', and 'mu'. It lists reflections 0, 1, 2, and 3.
- SimulatedModeComputedAngles:** A text area showing simulation results and error messages like 'Unobtainable reflection...'.
- Status Bar:** Displays a red error message: 'hkl calculations not available. Please configure the diffractometer Crystal not configuredReflection not configured'.

## 2D Detectors

- LIMA : Library for Image Acquisition
  - Lima is a C++ library for integrating 2D detectors developed at ESRF
  - Oriented to high-speed detectors
  - Separate hardware specific code from common software
  - Provide software alternatives to “missing” hardware capabilities
  - Plug-in architecture for detector integration
  - Web site : <http://www.lima.blissgarden.org>
  - Contributors: SOLEIL, PETRA-III, FRM-II, ALBA, RAYONIX, ADSC



## 2D Detectors

- **Geometric image transformations:**
  - Reconstruction of discontinuous readout
  - Bin, Rol, Flip & Rotation
  - Stripe concatenation
- **Basic image processing:**
  - Multi-Rol Statistics
  - Centroid (Beam Position Monitoring)
  - Rol Spectrum
  - Background subtraction, Flat-field normalisation
  - Spatial distortion correction
  - Frame accumulation
  - Image Mask

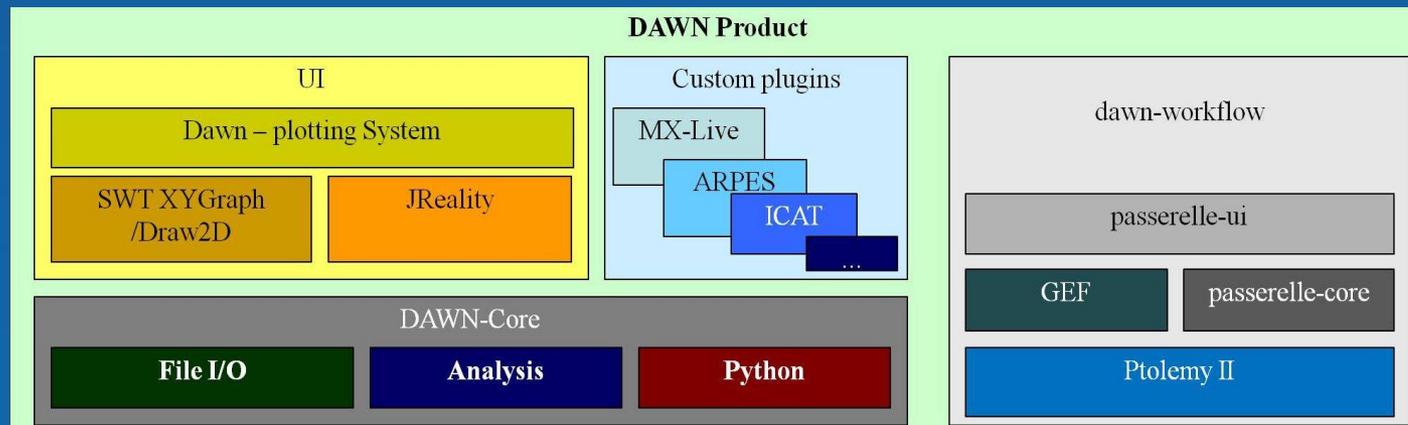
## 2D Detectors

- Available detector plug-ins:

Detector	Interface
Basler	GigE
Prosilica	GigE
Ueye (IDS)	GigE
Pilatus (Dectris)	300K, 1M, 2M, 6M, 6M-F
PCO Edge	Camera-Link
Pco Dimax	GigE + Camera-Link
PhotonicScience 4022	USB
RoperScientific	PVCAM SDK
Andor I-Kon	USB
PerkinElmer Flat-Panel	Proprietary board
ADSC 315r	Proprietary board
MarCcd 165	Proprietary board
Mythen (strip detector)	Proprietary board
XPAD	Proprietary board
Maxipix (ESRF)	Espia
Frelon 2K (ESRF)	Espia

## Data Analysis Workbench

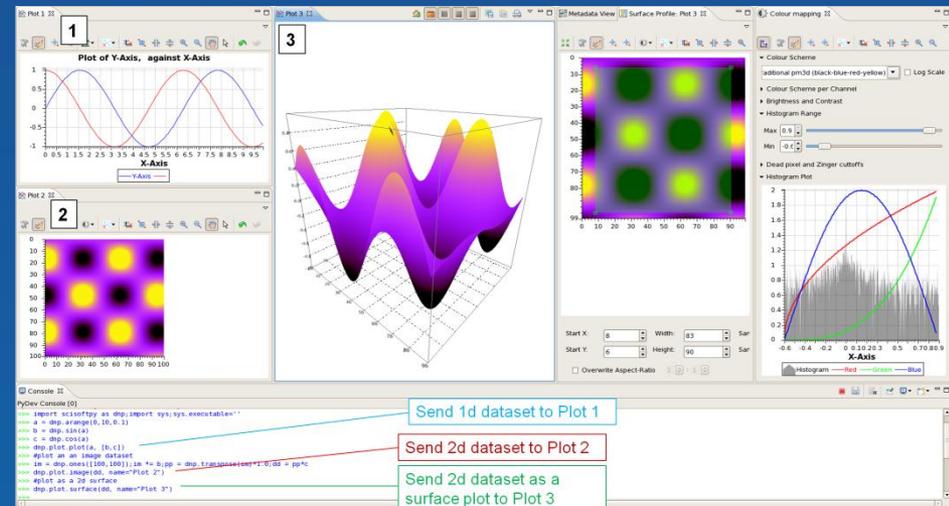
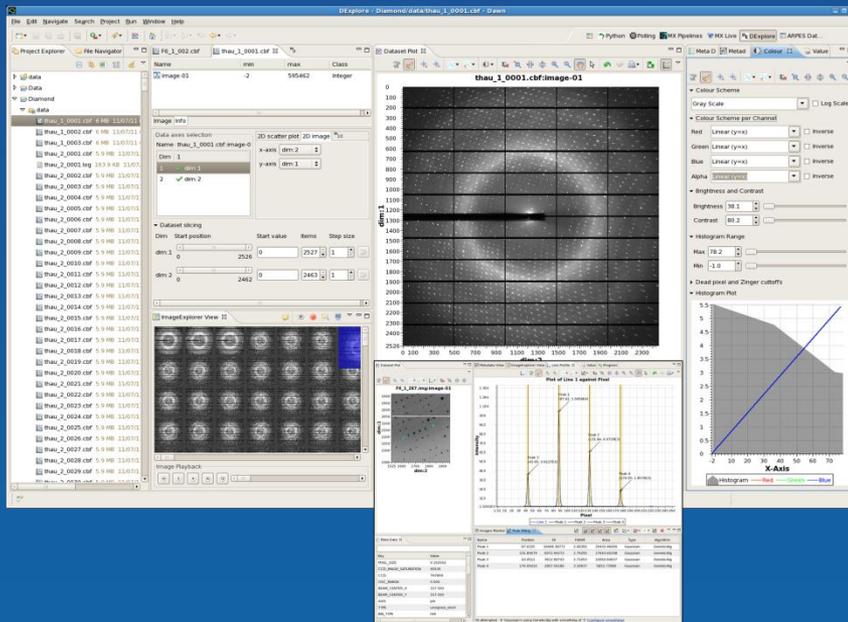
- Data analysis workbench with a workflow editor
  - Developed by DIAMOND and ESRF
  - Viewing scientific data: 1D, 2D and 3D datasets
  - Data exploring, data analysis and data saving
  - Importing and running other tools based on Eclipse RCP
  - Editing and running python scripts for data analysis
  - Web site : <http://www.dawnsci.org>



## Data Analysis Workbench

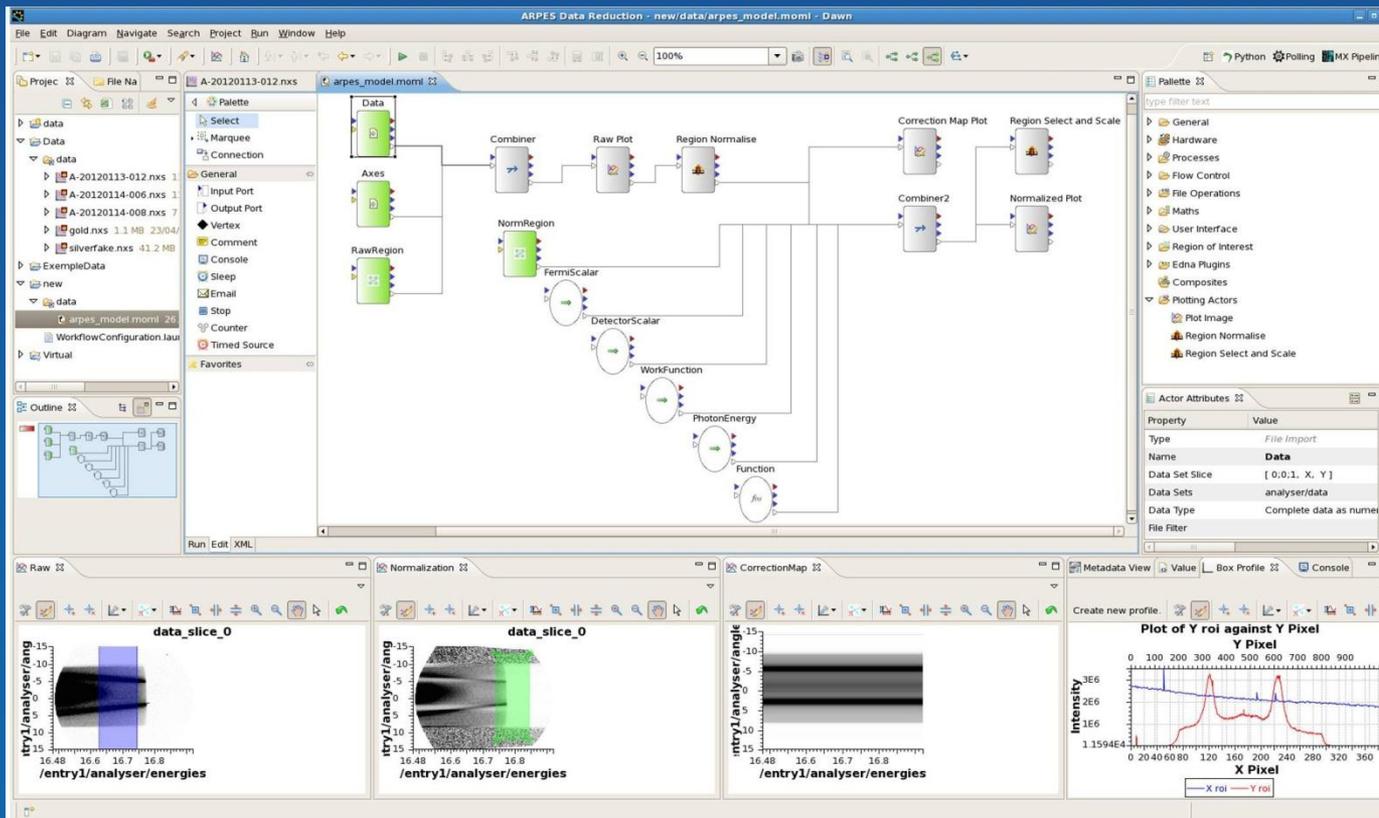
### Data Explorer Perspective

### Python Perspective



## Data Analysis Workbench

- Designing pipelines for data analysis
  - Workflow editor Passerelle
  - Customized with a library of data analysis actuators



## How to try it?

- Binary Packages
  - Available for Ubuntu (Debian) Linux in the standard distribution
  - Available for Windows on the Tango web site
- The Tango Box
  - A virtual Linux machine with most of the Tango packages installed and configured for easy testing
  - Runs with VMware Player
- Common Tango web site : <http://www.tango-controls.org>
- A mailing list for all questions and propositions to the community