Managing Tango Device Servers

Tango Workshop at ICALEPCS'23

Cape Town - South Africa

Sergi Rubio Manrique - ALBA Synchrotron

Setup of a Tango Development Environment

SKA Docker Images:

https://gitlab.com/ska-telescope/ska-tango-images

Those images are documented in the existing Tango Development Environment for SKA:

https://developer.skatelescope.org/en/latest/tools/tango-devenv-setup.html

Tango Box OVA Image:

A preinstalled linux virtual machine with all Tango Services instantiated:

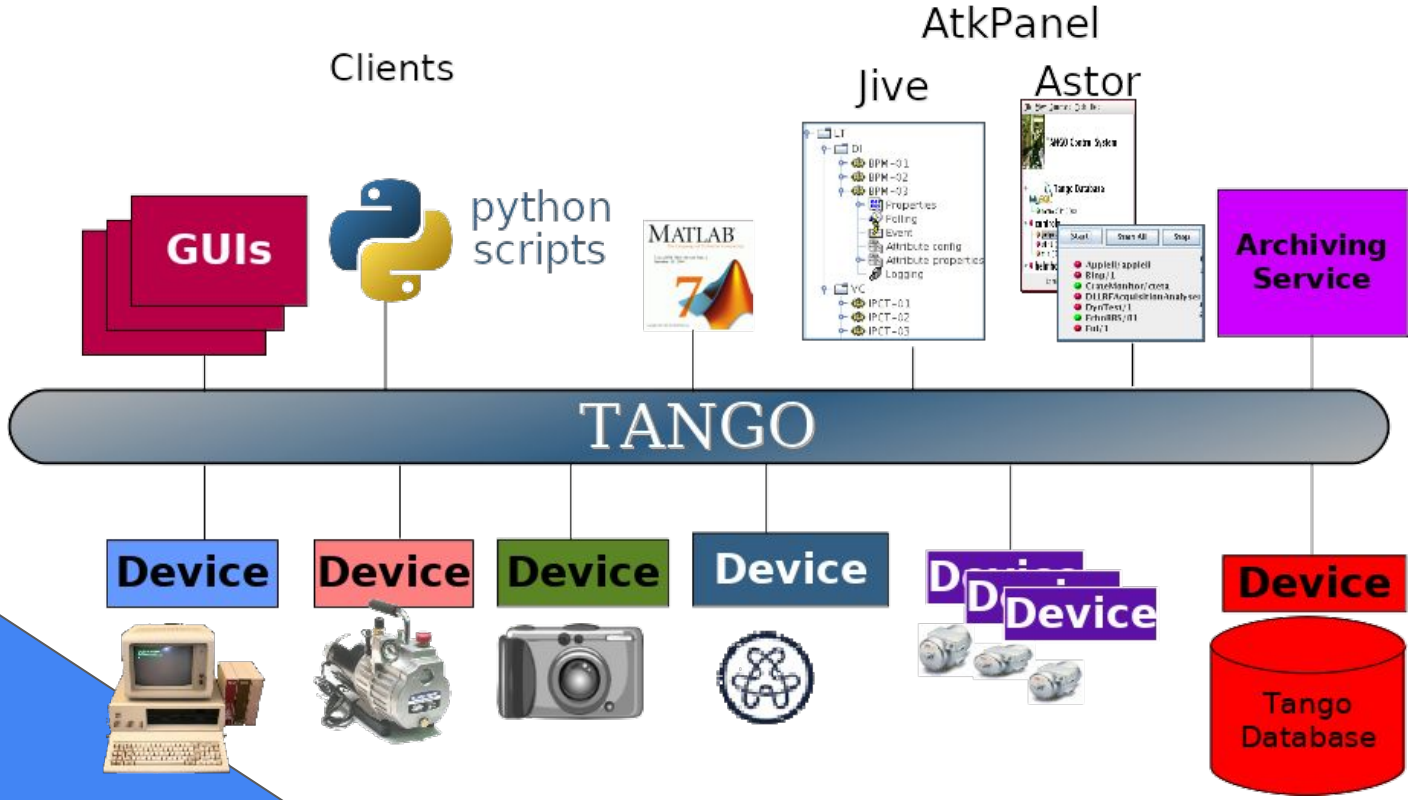
https://tango-controls.readthedocs.io/en/latest/installation/vm/tangobox-9.2.html

conda-forge

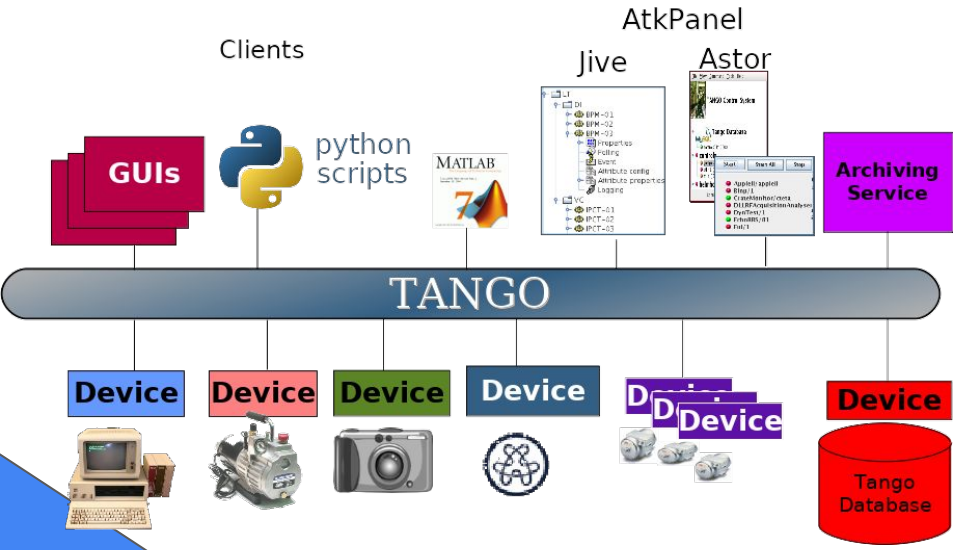
```
micromamba install -c conda-forge cpptango jive tango-astor pogo
tango-starter tango-test pytango taurus
```

https://beenje.github.io/blog/posts/developing-and-compiling-tango-with-conda/
https://pytango.readthedocs.io/en/latest/start.html

Tango Control System Overview



Tango Control System Overview



Everything in Tango is either a **Client or a Device** (including the database)

Database: Keeps servers, devices, properties, attributes configuration and host status

Device Servers: Each server is a process, in which several devices run instantiating Tango Device Classes plus an special Admin Device.

Device: Each instance of a Tango Device Class running within a device server.

Clients: locate devices thanks to the database, then instantiates proxies

Tango Devices mirror hardware devices

Ion Pump

Ion Pump

Ion Pump

DUAL Controller

Splitter Box

Serial Line

port0

port5

Network

5+ Tango devices

1 VacuumController instance

x2

VARIAN DUAL CONTROLLER

LIN

HV1

REM 1

SP

HV2

SR-VC-IPCT-RKA01A08-01

DB09M0

HV cable not bakable

CXL01M0

CHV01M0

ALBA VC SPLITTER BOX

HV INPUT

HVIN

IO2

HVO4

HVO5

SR-VC-SPBM-RKA01A08-01

CHV01M0

CHV01M0

CHV01M0

CHV01M0

HV cable Radiation resistant not bakable

HV cable Radiation resistant not bakable

HV Conector VARIAN DESIGN

HV Conector VARIAN DESIGN

HV Conector VARIAN DESIGN

HV Conector VARIAN DESIGN

Ion Pump

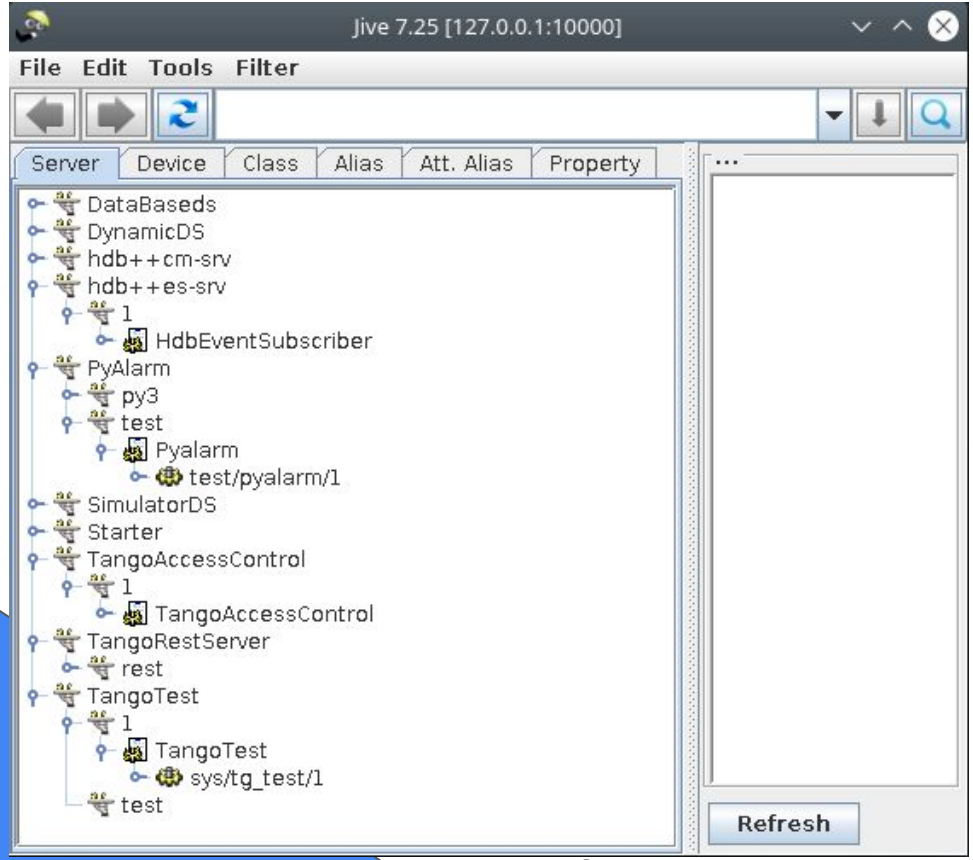
Ion Pump

Ion Pump

Ion Pump

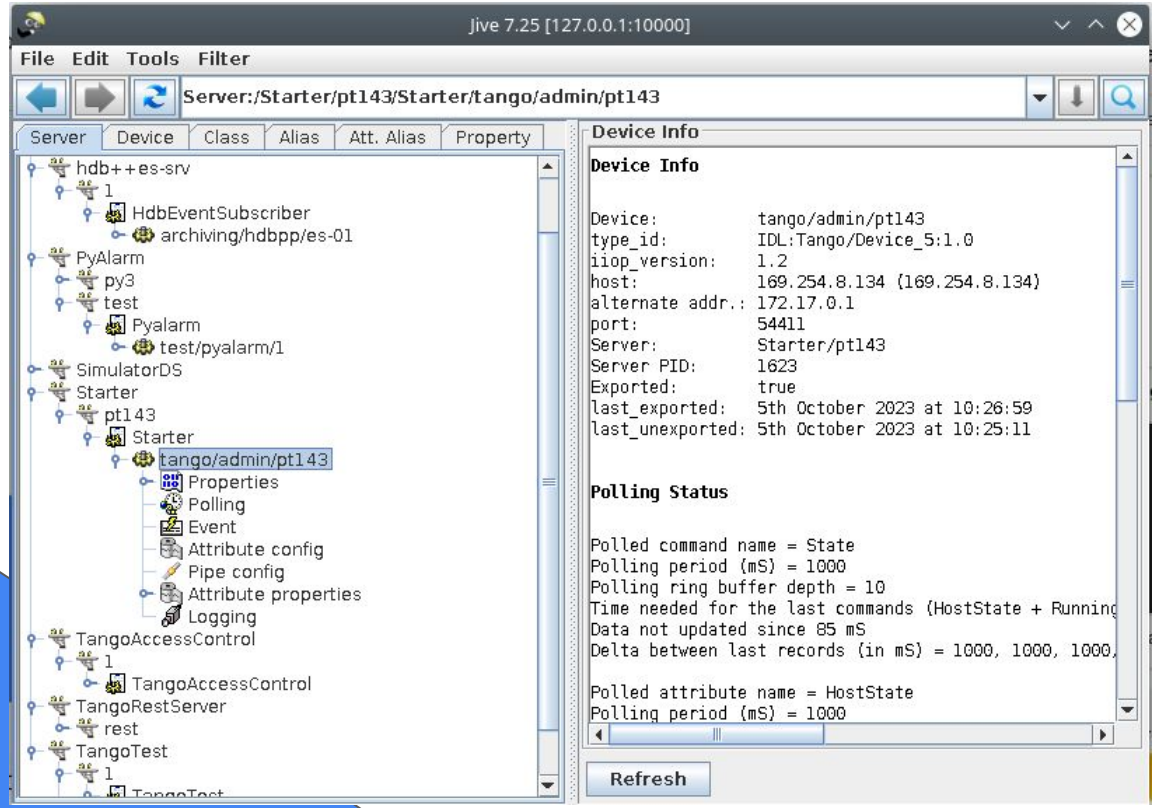
HVIN

Browsing the database with Jive



Jive allows to browse all existing devices (active or not) classified by its Server (process) name, Device name or Class name.

Browsing the database with Jive

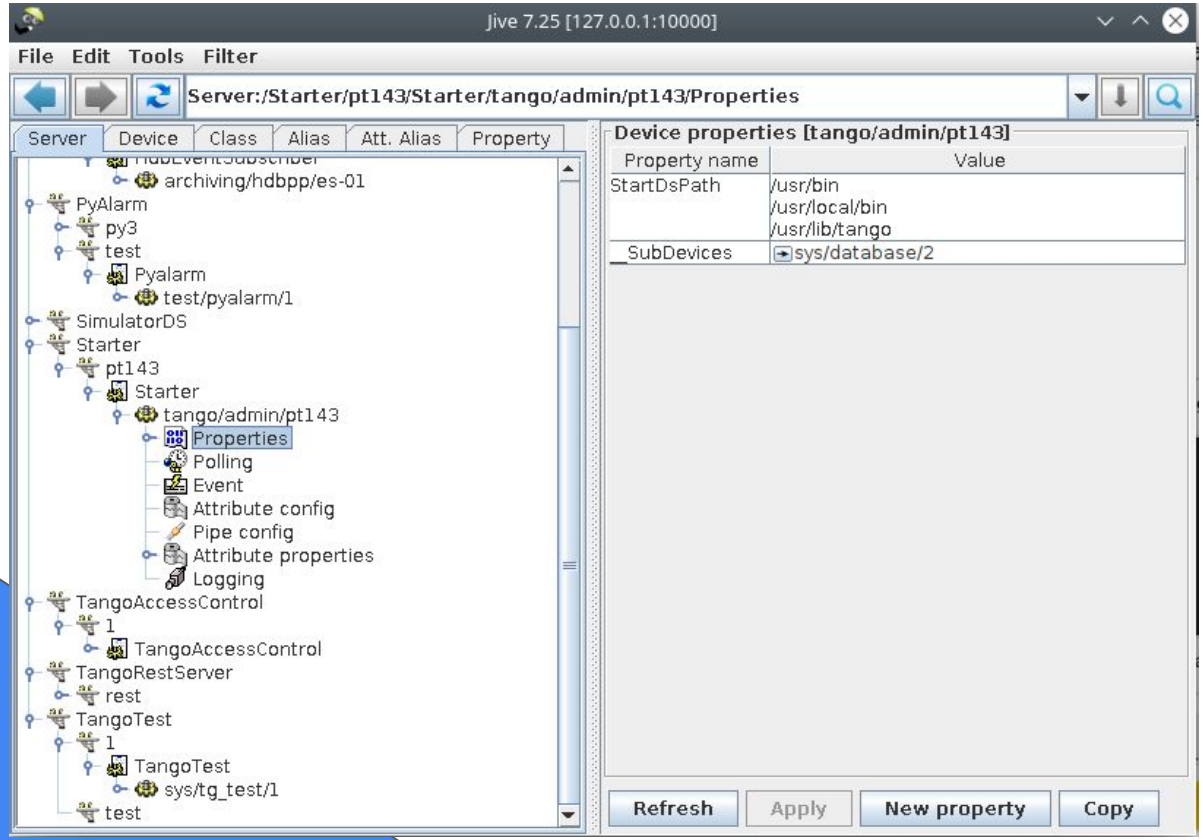


Jive displays all the required information regarding a device server.

This includes the host where it is running, and the configuration of all its attributes.

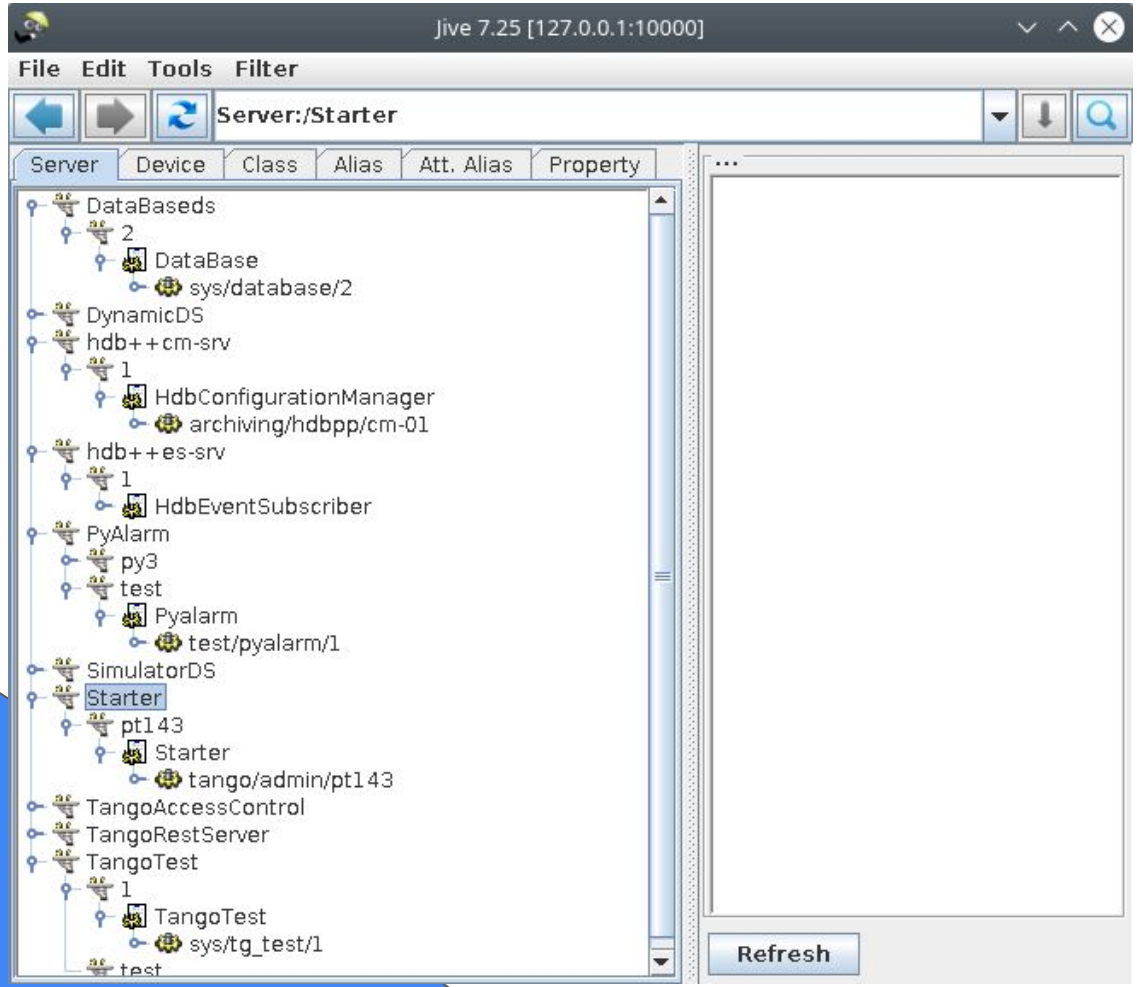
It provides all the functionality to create and configure devices.

Browsing the database with Jive



Properties allow to store custom configuration values for our devices.

Properties can exist at System, Device or Class levels.

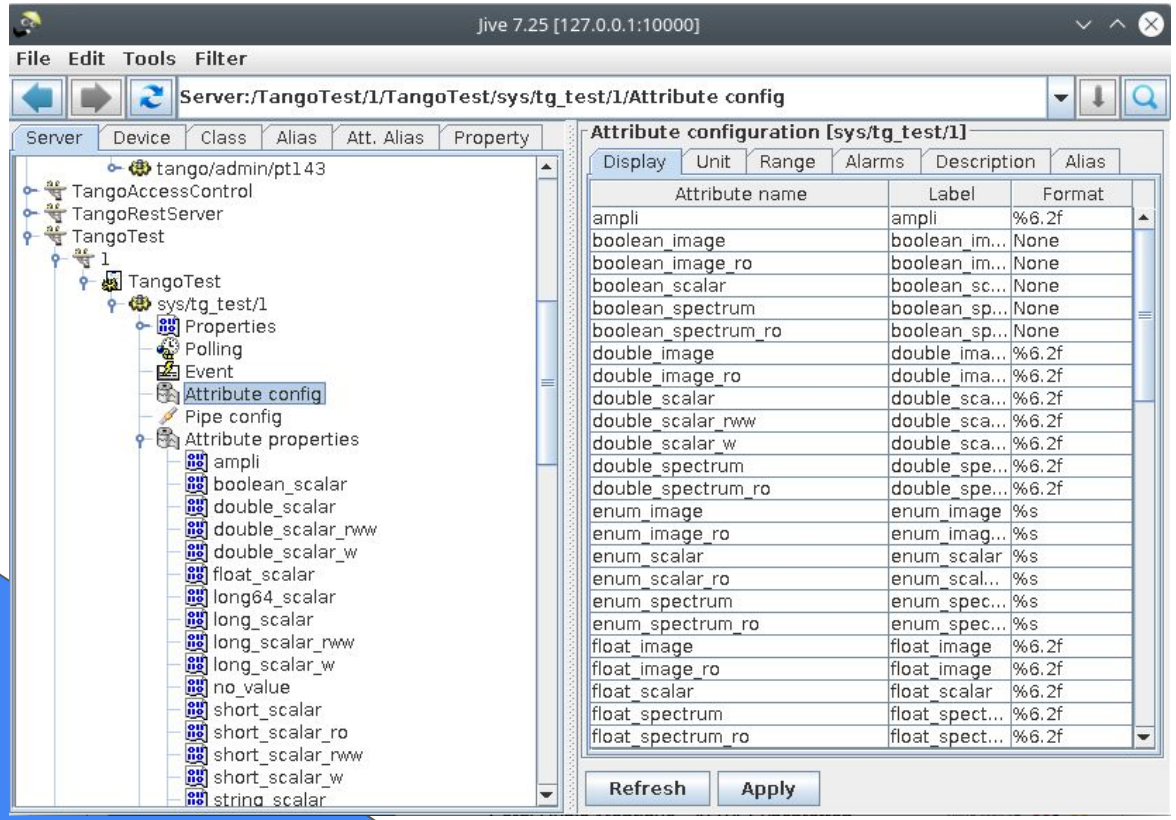


An "empty" control system already contains multiple devices, used by different Tango Services (database, archiving, alarms, ...)

The "Starter" device controls all servers running in a host.

Each server has an special "admin" device controlling its devices.

Browsing Attributes



For running devices, Jive allows to configure the attribute format and label (to be displayed by clients), the alarm levels, the event filters and the internal refresh polling amongst many other configuration parameters.

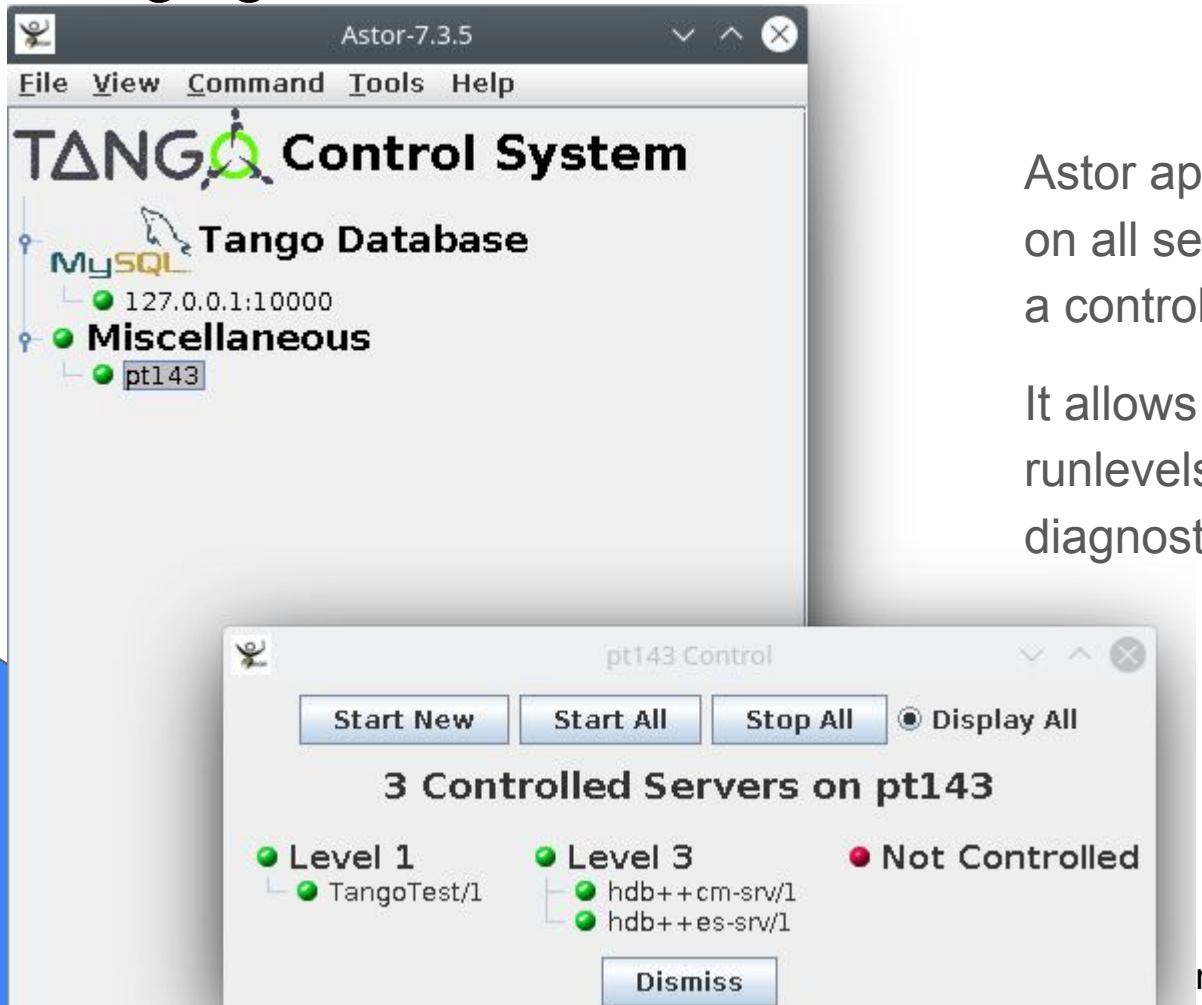
All these parameters can be set by code, but Jive allows to tune them to our specific needs.

Scripting: PyTango

```
In [1]: import PyTango
In [2]: db = PyTango.Database()
In [3]: db.get_device_property( 'tango/admin/pt143','StartDSPath')
Out[3]: {'StartDSPath': ['/usr/bin', '/usr/local/bin', '/usr/lib/tango']}

In [4]: dp = PyTango.DeviceProxy( 'tango/admin/pt143' )
In [5]: dp.info()
Out[5]: DeviceInfo(dev_class = 'Starter', dev_type = 'Uninitialised', doc_url = 'Doc
arter/pt143', server_version = 5)

In [6]: dp.get_attribute_list()
Out[6]: ['HostState', 'RunningServers', 'StoppedServers', 'Servers', 'State', 'Status

In [7]: dp.HostState
Out[7]: 0

In [8]: dp.RunningServers
Out[8]: ('hdb++cm-srv/1', 'hdb++es-srv/1', 'TangoTest/1')
```

brief demo ...

Managing servers and hosts: Astor



Astor application provides full control on all servers and devices running on a controls host.

It allows to start/stop devices, assign runlevels and execute testing and diagnostic tools.

Scripting: fandango

fandango provides Astor python API, providing the same functionality than astor tool.

```
pip3 install fandango
```

fandango can be used from python shell:

```
import fandango as fn

fn.tango.add_new_device('DynamicDS/1','DynamicDS','test/dyn/1')
astor = fn.Astor()
host = fn.linos.MyMachine().hostname
astor.start_servers('DynamicDS/1',host=host)
astor.set_server_level('DynamicDS/1',level=3,host=host)
```

methods from fandango can also be launched linux shell:

```
$: fandango add_new_device DynamicDS/1 DynamicDS test/dyn/1

$: fandango put_device_property test/dyn/1 DynamicAttributes "T=t%10"

$: tango_servers $HOSTNAME start DynamicDS/1
```

Events in Tango: polling vs. push

Event communication between devices and clients can be setup in 3 ways:

- programmatically: using push_*_event() methods in your code

- from pogo: enforcing the default event configuration, independent from implementation

- on runtime: using Jive or the Tango API to configure periodic polling and event filters in your device, that will trigger event on change

These 3 options are available for all attribute config parameters (e.g. qualities, alarm ranges)

Events in Tango: push on polling

Device polling [sys/tg_test/1]

Command | Attribute | Settings

Attribute name	Polled	Period (ms)
ampli	☐	
boolean_image	☐	
boolean_image_ro	☐	
boolean_scalar	☐	
boolean_spectrum	☐	
boolean_spectrum_ro	☐	
double_image	☐	
double_image_ro	☐	
double_scalar	☑	1000
double_scalar_rww	☐	
double_scalar_w	☐	
double_spectrum	☐	
double_spectrum_ro	☐	
enum_image	☐	
enum_image_ro	☐	
enum_scalar	☐	
enum_scalar_ro	☐	
enum_spectrum	☐	
enum_spectrum_ro	☐	

Event [sys/tg_test/1]

Change event | Archive event | Periodic event

Attribute name	Absolute	Relative
ampli	None	None
boolean_image	None	None
boolean_image_ro	None	None
boolean_scalar	None	None
boolean_spectrum	None	None
boolean_spectrum_ro	None	None
double_image	None	None
double_image_ro	None	None
double_scalar	1e-12	None
double_scalar_rww	None	None
double_scalar_w	None	None
double_spectrum	None	None

When using the default polling mechanism, values will be read periodically and compared against the event config.

Scripting: dsconfig

https://gitlab.com/MaxIV/lib-maxiv-dsconfig

This is a command line tool for managing configuration of Tango device servers. It runs on python 2.7 as well as 3.6 and up.

The goal of this project is to provide tools for configuring a Tango database in a convenient way. Right now the focus is on supporting Excel files as input ("xls2json"), but support for other formats should follow.

The main idea is that the input files are parsed and turned into an intermediate JSON format, specified by a schema. This file can then be given to the "json2tango" tool which then tries to make the database contents match, by adding, modifying or removing servers, devices and properties.

Run a device server without Database

Tango Control System can run devices without a database ... but still being able to use all Jive functionality to configure devices.

Device configuration can be exported from Jive once configured ... and then imported somewhere else (e.g. your Raspberry Py) from a device server on runtime.

```
myserver myinstance_name -file=/tmp/MyServerFile -ORBendPoint giop:tcp::<port number>
```

More info:

https://tango-controls.readthedocs.io/en/latest/administration/deployment/without-sql-db.html#example-of-device-server-started-without-database-usage

https://tango-controls.readthedocs.io/en/latest/administration/deployment/starting.html#without-database

Other tools

The Tango ecosystem is huge, multiple archiving backends, multiple ways to configure the database, multiple gateway devices and façades, python and c++ alarm systems.

Most of services are based on devices, configured with properties and can be easily configured using dsconfig or fandango.

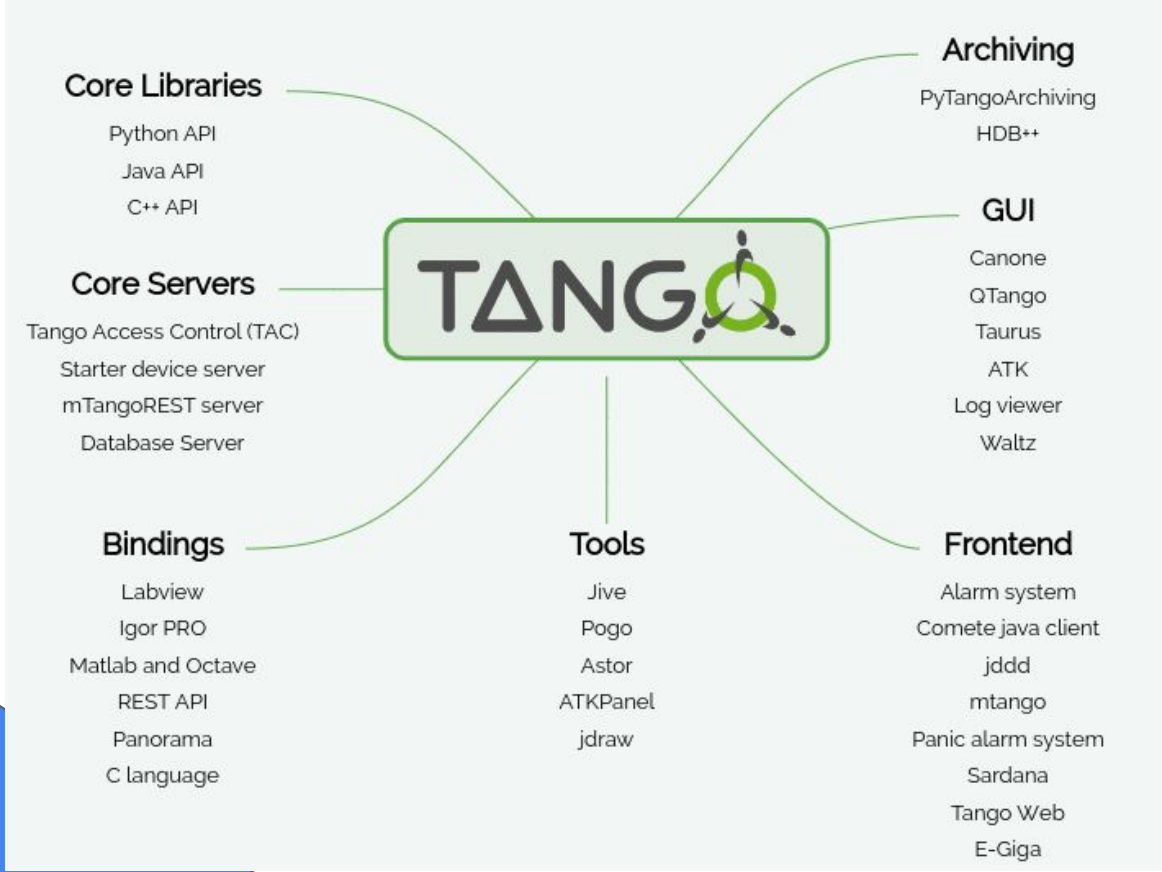
Community projects are hosted at:

https://gitlab.com/tango-controls/

And there's a huge catalogue of existing software/hardware Tango Classes:

https://www.tango-controls.org/developers/dsc/

Tango Ecosystem



Conclusions / Questions

https://tango-controls.readthedocs.io/en/latest/installation/vm/tangobox-9.2.html

- https://tango-controls.org
- https://www.tango-controls.org/community/forum/
- https://tango-controls.readthedocs.io/
- https://pytango.readthedocs.io/
- https://gitlab.com/tango-controls
- https://gitlab.com/tango-controls/fandango
- https://gitlab.com/MaxIV/lib-maxiv-dsconfig