TANGO Status at Ego May 2014.

Martin Mohan.

VIRGO/EG0

23 May 2014

Change Record

Version	Date	Section Affected	Reason / Remarks
V1R0	23 May 2014	All	DRAFT
V1R1			.

Table of Contents

Table of Figures

Related Documents

[1] Cryotrap Control With Tango Friday, 27 April, 2012
https://tds.ego-gw.it/ql/?c=8978

[2] TANGO Software testing at Schio on the Cryotraps (Thursday, 10 April, 2014)
https://tds.ego-gw.it/ql/?c=10196

Institute Acronyms

EGO:	European Gravitational Observatory, Pisa, Italy
ESRF:	European Synchrotron Radiation Facility
LAL:	Laboratoire de l’Accelerateur Lineaire, Paris (Orsay), France
Nikhef:	National Institute for Nuclear and High energy physics,Amsterdam, NL

1 Introduction

1.1 Tango introduction

This is the report on current state of Tango at Ego in May 2014.
Tango was adopted by vacuum group in 2011 to replace custom software at Ego and was also subsequently adopted by the SAT subsystem. It is also used for some INJ software.
Currently several Tango servers are tested and installed at Ego. See Figure 1for overview.

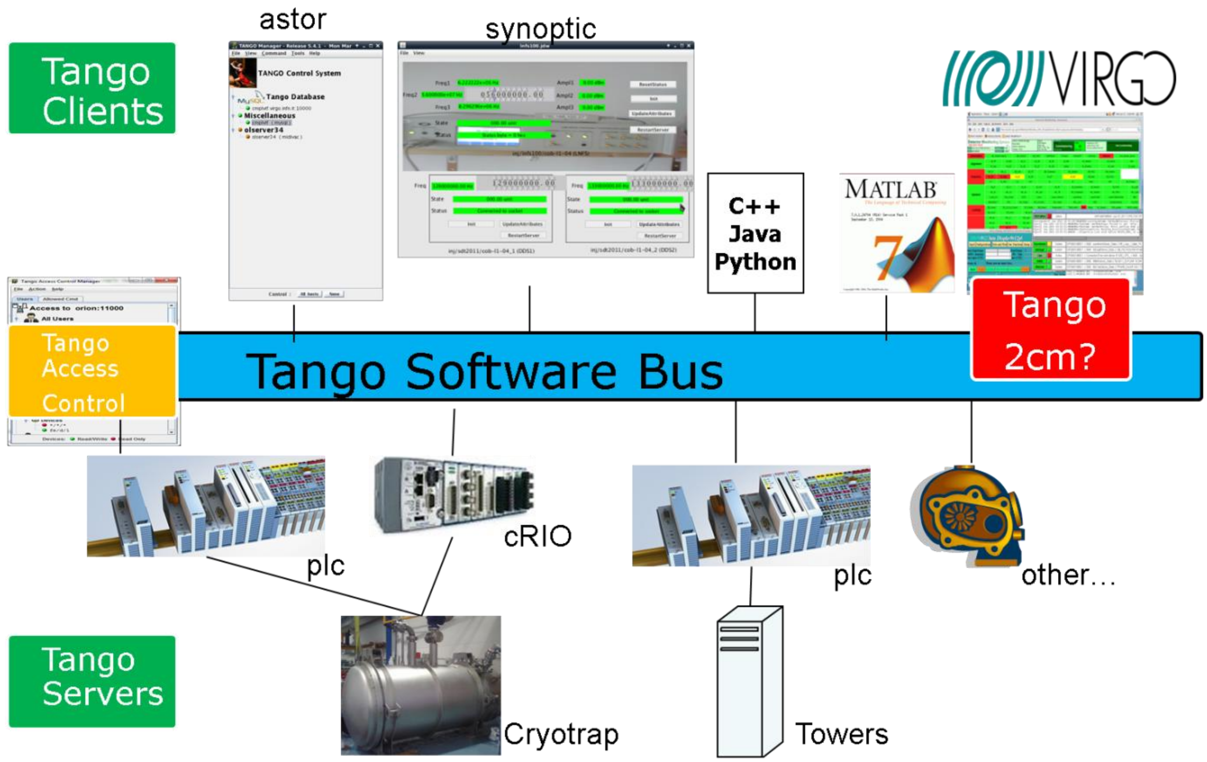


Figure 1 Overview of Tango Clients and Servers

1.2 Vacuum system and Tango

Tango was used to test the Cryotraps at Ego. First software to test the Cryotraps using Tango was created in April 2012[1].Since then Tango was used to successfully test Cryotrap prototype at Schio and in Final tests at Schio in April 2014[2].

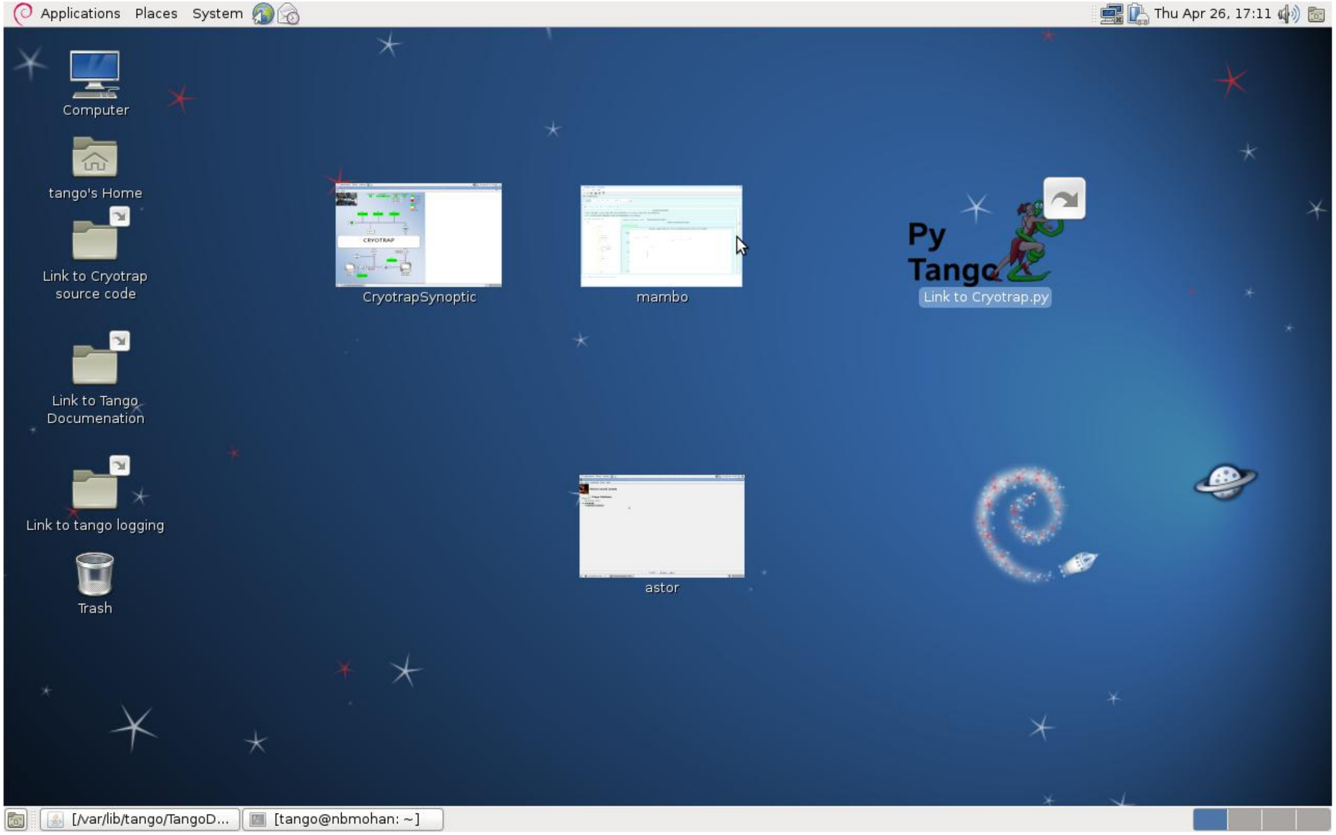


Figure 2 Screen dump of laptop used to test Cryotraps at Schio

1.3 SAT/INJ system and Tango

SAT subsystem is also using the Tango system and some small INJ drivers also use Tango.

2 Tango Installation and Linux

Initially it was attempted to put Tango on Scientific Linux which is the default operating system used at Ego. This proved difficult due to many software conflicts so Tango was installed on Debian linux machines for which Tango is specifically packaged.

.

2.1 Tango and Scientific Linux at Virgo/Ego.

Scientific Linux http://en.wikipedia.org/wiki/Scientific_Linux

Virgo/Ego uses Scientific Linux as the main operating system. Although Tango can be installed on Scientific linux it can be quite difficult and time consuming because Tango is not packaged for this operating system.

At Ego the Scientific Linux machines are mounted together on disks with names like /virgoDev and /virgoApo and developers add their programs to these disks. For a large system like Tango which supports many libraries, having all software installed together can lead to conflicts of different software versions. E.g. the version of python installed on /virgoXXX can be different from that used by Tango.

Ref: http://sprserver.ego-gw.it/mantisbt/view.php?id=1151

Despite these difficulties some Tango software servers have been installed on Scientific Linux at Ego, although a Debian linux machine is still needed for the configuration of these servers.

2.2 Tango and Debian Linux at Virgo/Ego.

http://en.wikipedia.org/wiki/Debian

Debian is seen as a stable Linux distribution and has roughly a 2 year release cycle. Debian is the operating system used at ESRF (which originally developed Tango) and most of the Tango software is packaged for Debian using the Debian aptitude package manager. This means that software conflicts are resolved by the Tango and Debian communities. Aptitude packaging makes it easies to install, maintain and upgrade the software and having a more standard installation means better support can be provided by the Tango community.

Tango uses cobra and zeromq to communicate between different servers. Access to different Tango buses is achieved by setting the environment variable TANGO_HOST.

The advantages of using Debian linux in terms of reliability and support outweigh those of using Scientific Linux.

3 Tango Servers

Tango servers are used to access hardware and are shown on the lower half of Figure 1.
Tango servers read/write hardware data from/to the Tango Software bus.
All servers on a Tango bus can be viewed and tested using the Astor client. Several servers have been written and tested. Some are shown on the lower half of Figure 1.
A Tango bus is selected by setting the environment variable TANGO_HOST.
The host “tangoserver” is used to test hosts for Vacuum equipment by LAL and Nikhef.
The host “cmplvtf” runs several other Tango servers used by INJ and vacuum.
Astor has also been installed both on Debian and Scientific Linux.
e.g. To see the hosts running on cmplvtf
On Scientific linux (csh): setenv TANGO_HOST cmplvtf:10000
On Debian linux   (bash):export TANGO_HOST=cmplvtf:10000

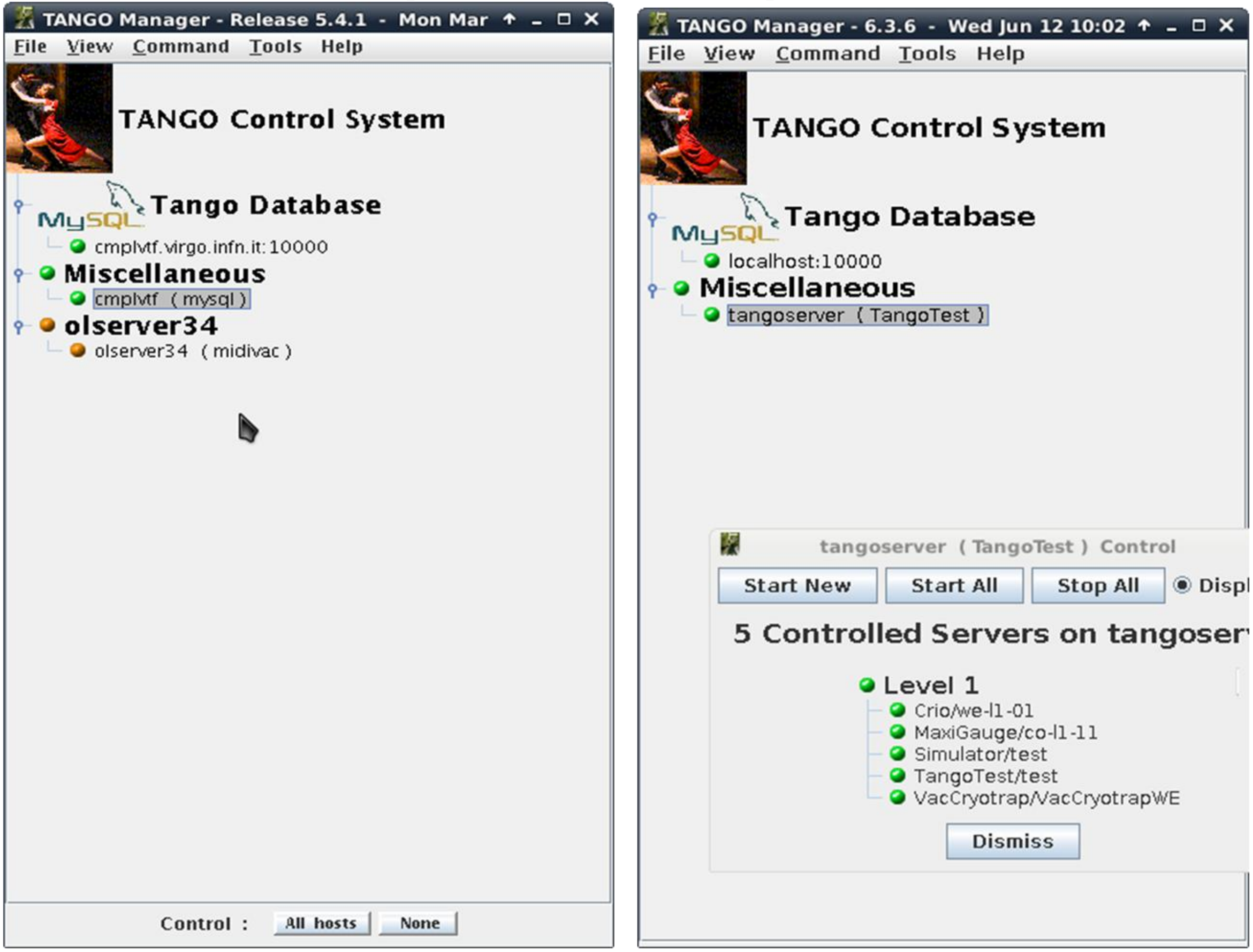


Figure 3 Astor running on hosts: Tangoserver cmplvtf.

.

3.1 Vacuum Servers

3.1.1 Crio:

A rack to control Nitrogen levels for the Cryotrap was developed at Nikhef. This uses modbus protocol for hardware communication. The first rack has been installed in the West End Building.
The Crio controller and a local interface were developed by Fred Schimmel (Nikhef). A Tango Modbus interface to Crio was developed by Martin Mohan (EGO). Network problems meant remote tests with Tango were not possible yet but preliminary local tests using Tango were partially successful.

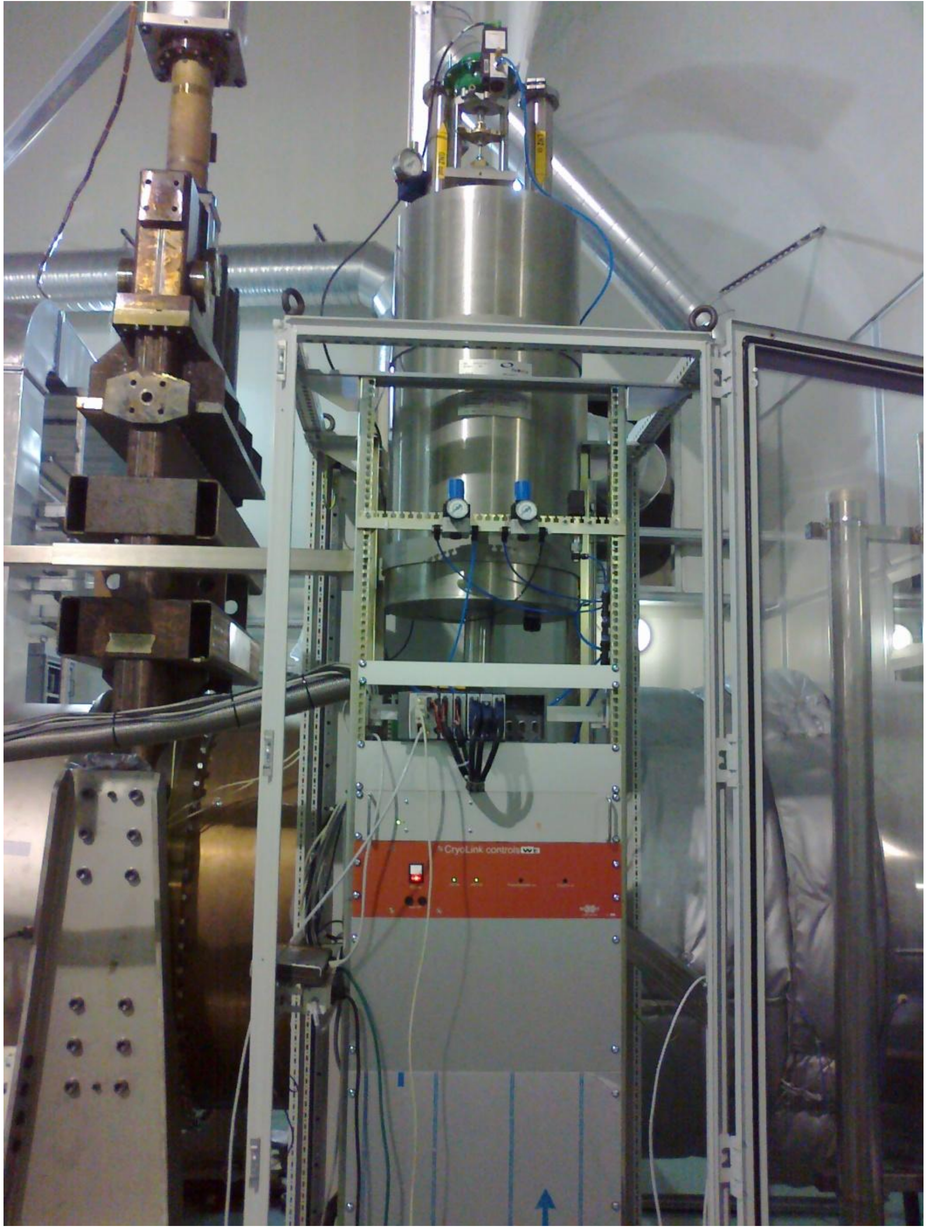


Figure 4 Nikhef Crio controller

3.1.2 VacPLC:

A rack to control pressure pumps and valves for the Cryotraps was developed at LAL. This uses modbus protocol for hardware communication. The first rack has been installed in the West End Building.

A local interface and A Tango Modbus interface to this rack was developed by Eric Jules (LAL).Preliminary tests have been carried out successfully both locally and remotely.

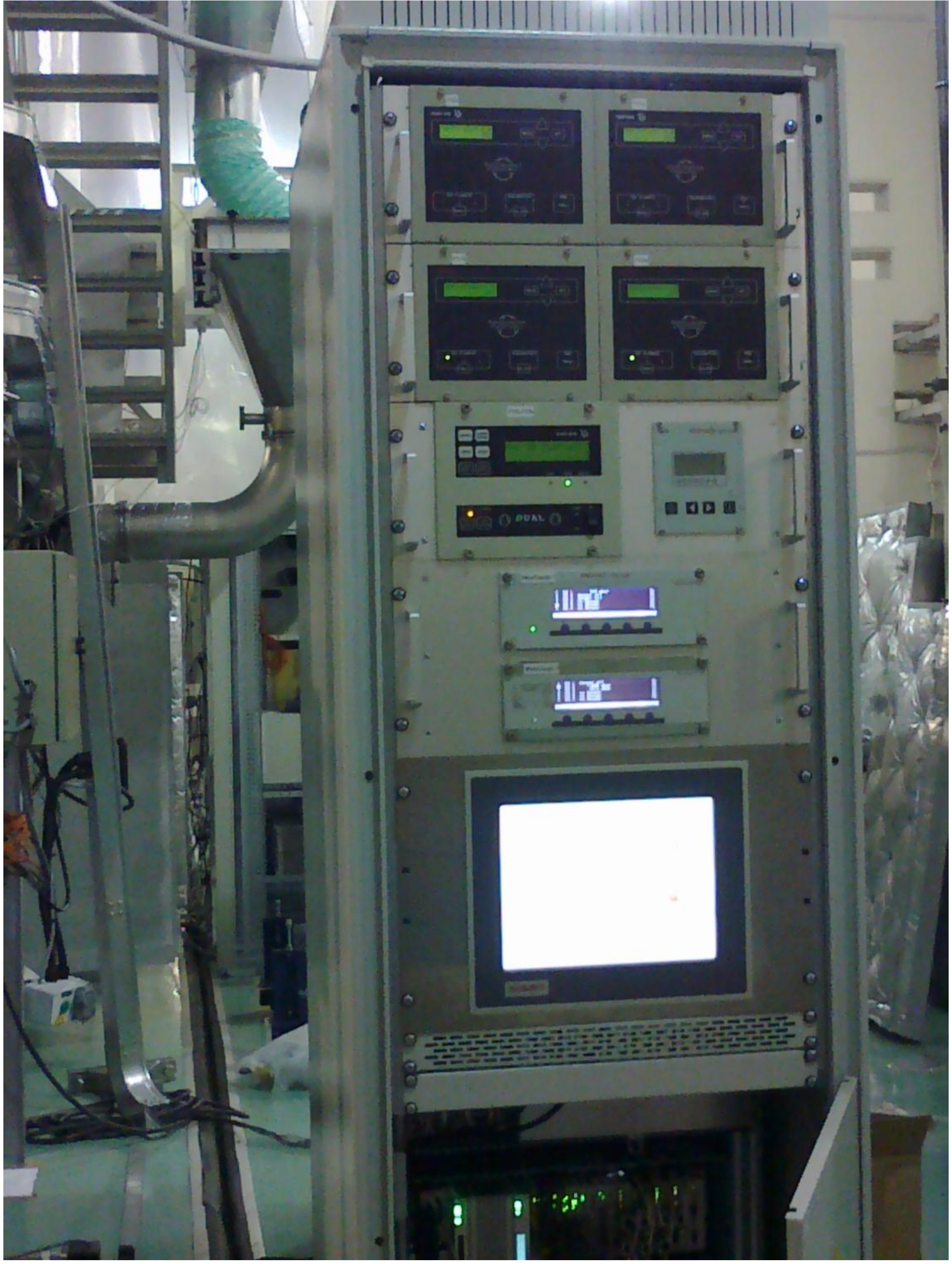
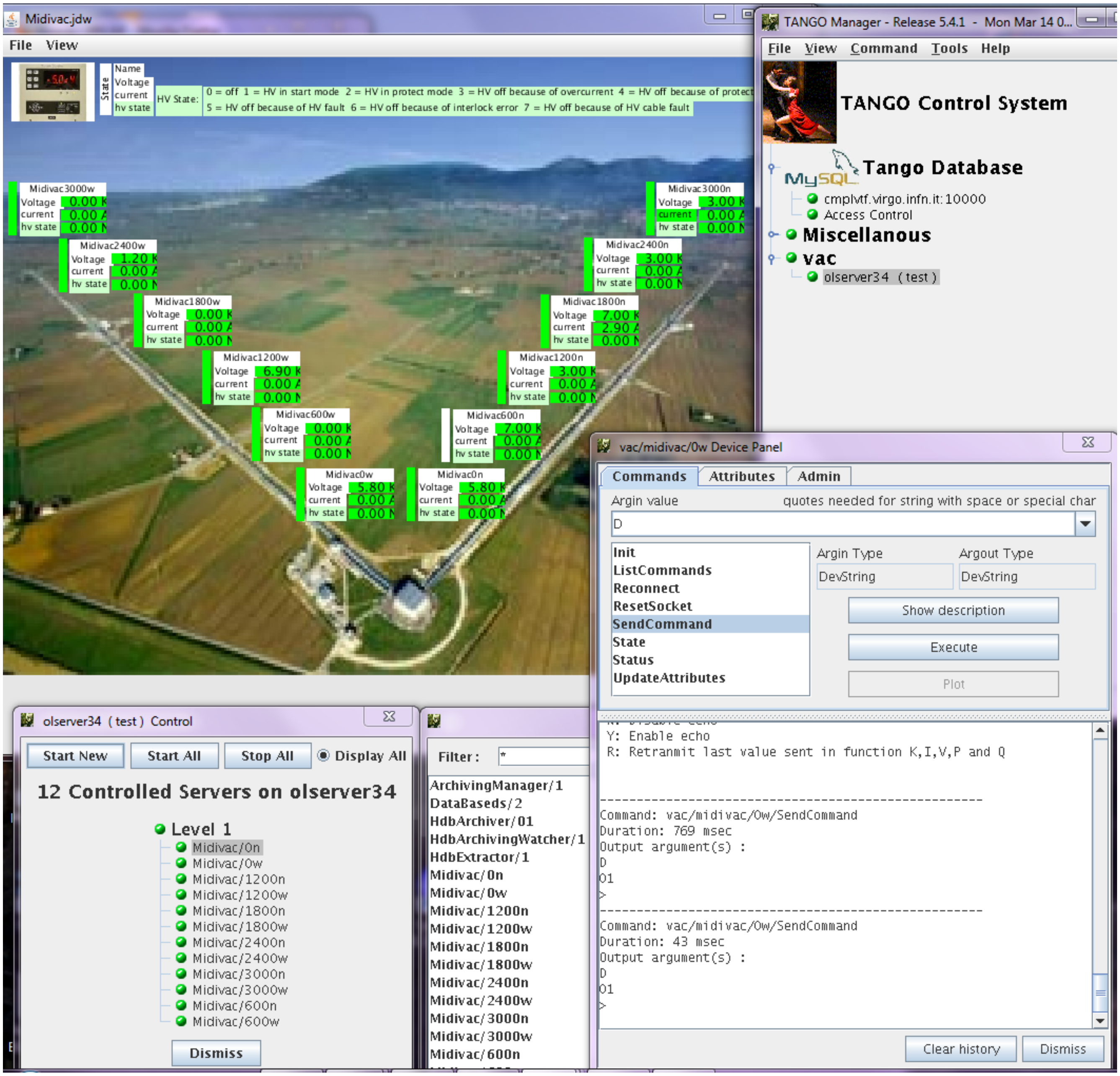


Figure 5 LAL Cryotrap control rack

3.1.3 Midivac:

12 Tango servers control the Midivacs. These have been installed and running since Friday 04 May 2012
https://tds.ego-gw.it/itf/osl_virgo/index.php?callRep=30748
The Tango servers run on Scientific Linux (olserver34) but the Tango use a Debian server (cmplvtf) for Tango configuration data.



Midivac.jdw
File View
State
Name
Voltage
current
hv state
HV State:
Midivac3000w Voltage 0.00 K current 0.00 A hv state 0.00 N
Midivac2400w Voltage 1.20 K current 0.00 A hv state 0.00 N
Midivac1800w Voltage 0.00 K current 0.00 A hv state 0.00 N
Midivac1200w Voltage 6.90 K current 0.00 A hv state 0.00 N
Midivac600w Voltage 0.00 K current 0.00 A hv state 0.00 N
Midivac0w Voltage 5.80 K current 0.00 A hv state 0.00 N
Midivac3000n Voltage 3.00 K current 0.00 A hv state 0.00 N
Midivac2400n Voltage 3.00 K current 0.00 A hv state 0.00 N
Midivac1800n Voltage 7.00 K current 2.90 A hv state 0.00 N
Midivac1200n Voltage 3.00 K current 0.00 A hv state 0.00 N
Midivac600n Voltage 7.00 K current 0.00 A hv state 0.00 N
Midivac0n Voltage 5.80 K current 0.00 A hv state 0.00 N
TANGO Manager - Release 5.4.1 - Mon Mar 14 0...
File View Command Tools Help
TANGO Control System
MySQL
Tango Database
cmplvtf.virgo.infn.it:10000
Access Control
Miscellanous
vac
olserver34 (test)
vac/midivac/0w Device Panel
Commands Attributes Admin
Show description
Execute
Plot
Clear history
Dismiss
olserver34 (test) Control
Start New
Start All
Stop All
Display All
Level 1
Filter :

0 = off 1 = HV in start mode 2 = HV in protect mode 3 = HV off because of overcurrent 4 = HV off because of protect
5 = HV off because of HV fault 6 = HV off because of interlock error 7 = HV off because of HV cable fault

Argin value quotes needed for string with space or special char

D

Init
ListCommands
Reconnect
ResetSocket
SendCommand
State
Status
UpdateAttributes

Argin Type: DevString
Argout Type: DevString

```
Y: Enable echo
R: Retranmit last value sent in function K,I,V,P and Q


----------------------------------------------------
Command: vac/midivac/0w/SendCommand
Duration: 769 msec
Output argument(s) :
D
01
>
----------------------------------------------------
Command: vac/midivac/0w/SendCommand
Duration: 43 msec
Output argument(s) :
D
01
>
```

12 Controlled Servers on olserver34

Level 1
- Midivac/0n
- Midivac/0w
- Midivac/1200n
- Midivac/1200w
- Midivac/1800n
- Midivac/1800w
- Midivac/2400n
- Midivac/2400w
- Midivac/3000n
- Midivac/3000w
- Midivac/600n
- Midivac/600w

Dismiss

Filter : *

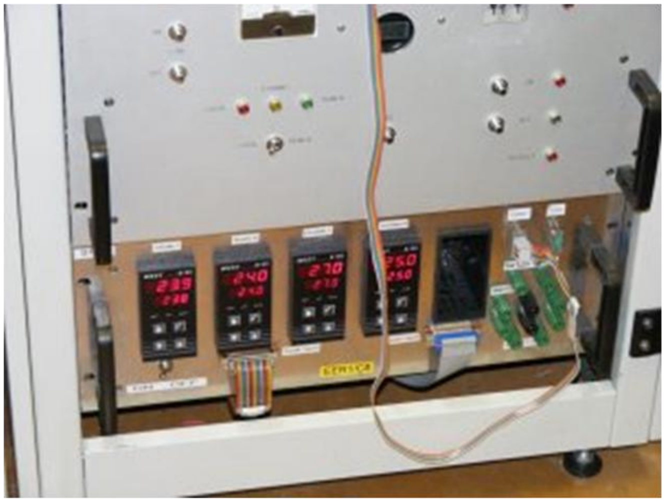
ArchivingManager/1
DataBaseds/2
HdbArchiver/01
HdbArchivingWatcher/1
HdbExtractor/1
Midivac/0n
Midivac/0w
Midivac/1200n
Midivac/1200w
Midivac/1800n
Midivac/1800w
Midivac/2400n
Midivac/2400w
Midivac/3000n
Midivac/3000w
Midivac/600n

Figure 6 Midivac

3.2 INJ Servers

3.2.1 LaserChiller/LaserChillDiodes:

These servers control the LaserChiller/LaserChillerDiodes. These Tango servers have been running for a while and they interact with cm software through a simple C++ interface.



3.2.2 Lnfs100:

This server along with sdt2011 below is used for frequency control by the injection system.

Figure 7 Lnfs100

3.2.3 sdt2011:

These are frequency generators used by injection and associate with lnfs100 above.

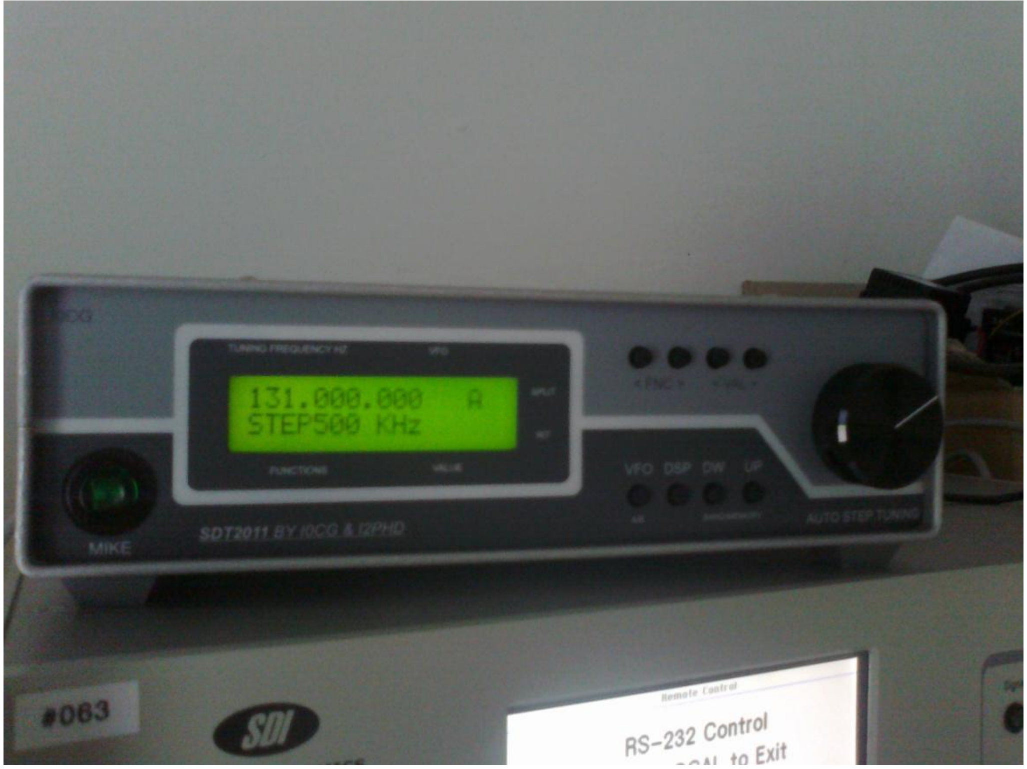


Figure 8 sdt2011

3.3 SAT Servers

Info can be provided by Giulio Ballardin (EGO).

4 Tango Clients

Tango clients are used to control the Tango Servers and are shown on the upper half of Figure 1. Tango Clients read/write data from/to the Tango Software bus.
Although all Tango Servers can be controlled with Tango’s Astor client many other clients are also available. Over the years the Tango community has developed clients such as GUI builders; language interfaces (C++, java and python) and interfaces to 3rd party clients such as Matlab.
See https://www.Tango-controls.org/clients/Tango-clients . Several of these clients have been tested at Ego.

4.1 Gui clients

The most obvious interface to a device server is a GUI. Tango provides several clients to create GUIs and one of the most intuitive is probably jdraw Figure 9

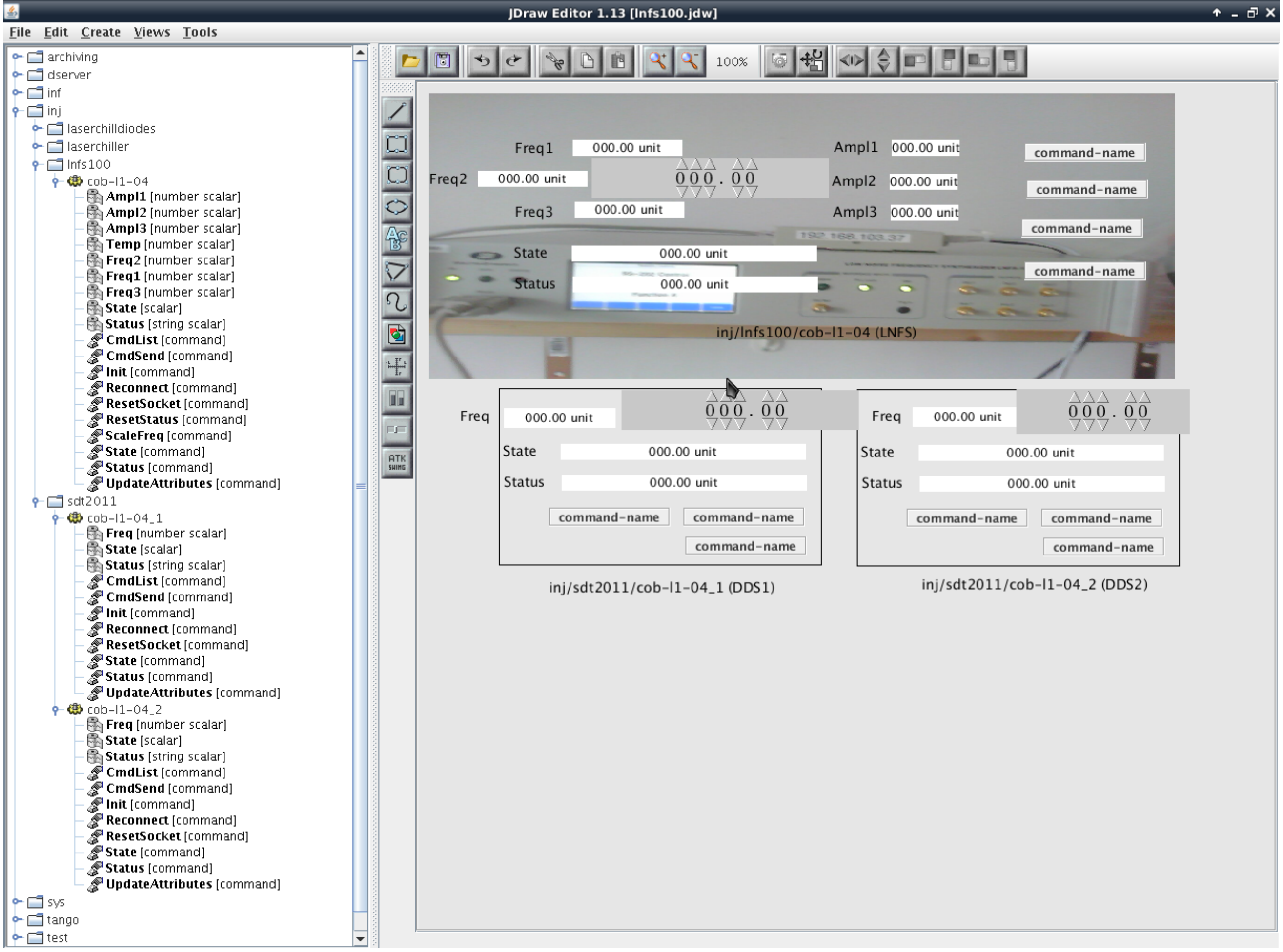
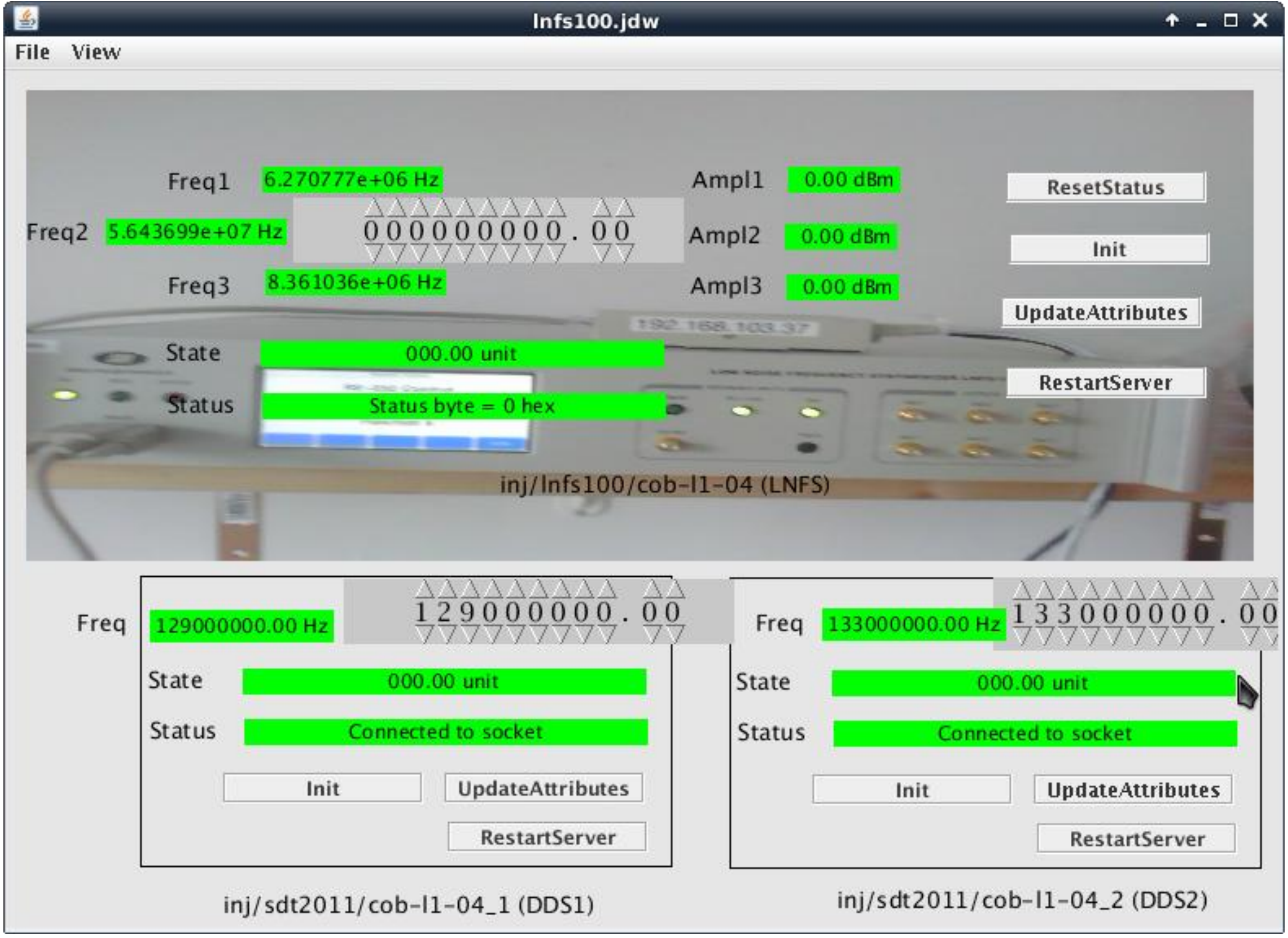


Figure 9 Lnfs GUI design - jdraw

A designer can insert pictures labels and diagrams using the jdraw editor. A pull down menu on the left hand side contains a list of attributes/commands which the designer then associates with the design.
Commands such as “RestartServer” and attributes such as Freq1,Freq2,Freq3 can then be dragged into drawing. In this way a fully functioning GUI can be created in a short time.

After saving the GUI can be launched using the client synopticAppli (see below).



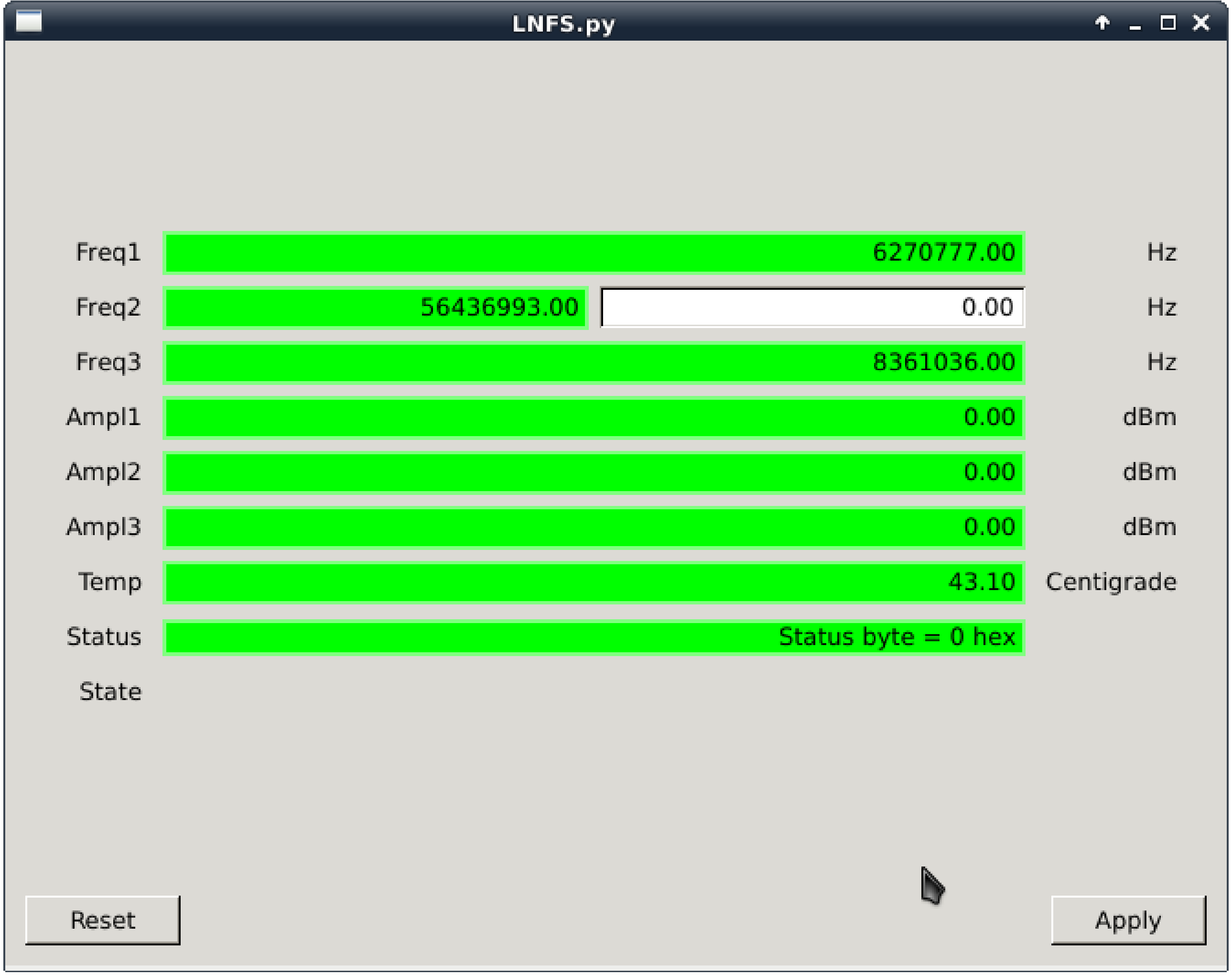
Freq1 6.270777e+06 Hz
Freq2 5.643699e+07 Hz
0000000000.00
Freq3 8.361036e+06 Hz
Ampl1 0.00 dBm
Ampl2 0.00 dBm
Ampl3 0.00 dBm
State 000.00 unit
Status Status byte = 0 hex
inj/lnfs100/cob-l1-04 (LNFS)

Freq 129000000.00 Hz
129000000.00
State 000.00 unit
Status Connected to socket
inj/sdt2011/cob-l1-04_1 (DDS1)

Freq 133000000.00 Hz
133000000.00
State 000.00 unit
Status Connected to socket
inj/sdt2011/cob-l1-04_2 (DDS2)

Figure 10 LNFS synoptic – created with jdraw

Several clients in addition to jdraw are available to create GUI's, such as jddd, Sardana and Taurus. Below is an example of another GUI for the same Lnfs server above created using Taurus client.



Freq1 6270777.00 Hz
Freq2 56436993.00 0.00 Hz
Freq3 8361036.00 Hz
Ampl1 0.00 dBm
Ampl2 0.00 dBm
Ampl3 0.00 dBm
Temp 43.10 Centigrade
Status Status byte = 0 hex
State

Figure 11 LNFS synoptic – created with Taurus

4.1.1 Vacuum Supervisor

The vacuum supervisor will be used to oversee all the Tango Servers. Building the vacuum supervisor will depend on the GUI client selected. Lower level security should depend on setting up TangoAccessControl (5.2).There are several other issues such as locking ownership which must also be addressed.

4.2 Cm2Tango client

Figure 1shows a client called cm2Tango. This represents a client which can be used to read/write from/to a Tango bus and convert the data into a format which is compatible with the Virgo client cm.

Cm is a Virgo command line interface used for slow data monitoring. If a cm interface to the Tango bus is created the Tango bus can be monitored using the current Ego/virgo cm clients.

Cm operates on Scientific linux and is written in C++. A C++ interface to the Tango bus has been installed on Scientific Linux providing the possibility of writing a cm2Tango interface

Successful tests have been carried out to read simple individual Tango servers into cm format. However this is not a sustainable solution as the Tango source code must be modified for every change or addition to a Tango server.

A generic cm2Tango interface should allow integration with existing Virgo clients.

4.3 Command Line clients

Tango provides Python, Java and C++ interfaces to the Tango bus. These allow automation of common tasks.

5 Tango in a Production environment

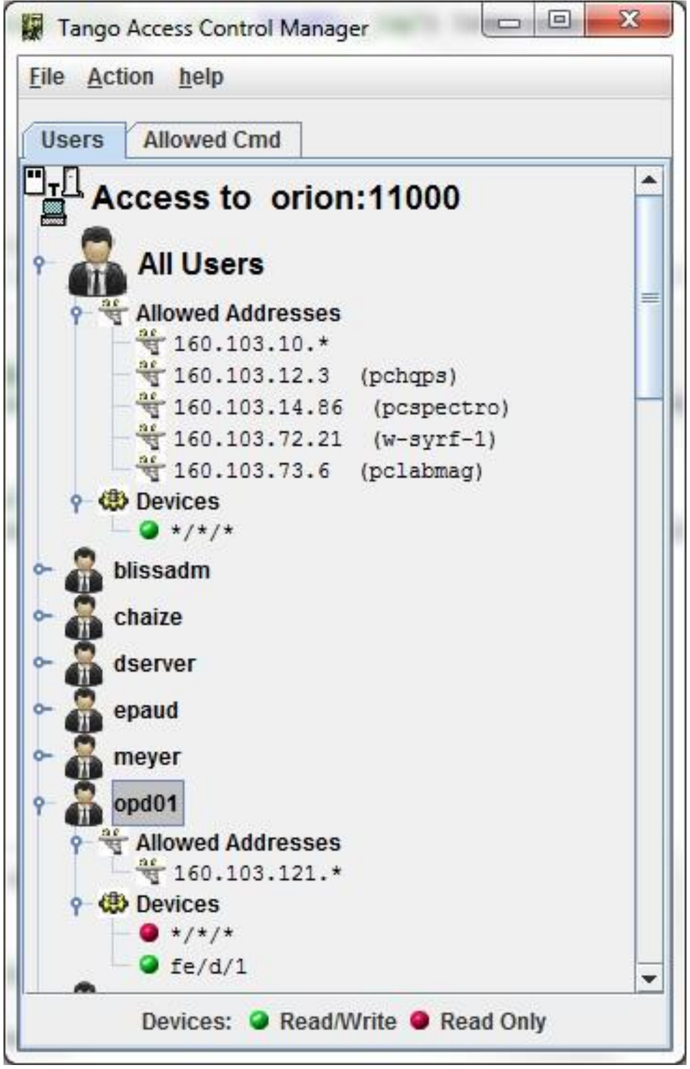
Several Tango test servers have been developed in a test environment.
For a production environment a common naming convention should be adopted and access control agreed upon among other things.

5.1 Tango Naming convention

Each Tango server must have a unique name. A name consists of 3 parts. A common naming common naming convention is still to be adopted by Ego.
An example of a suggested naming convention is:
Crio/we-l1-01 (Server Name/Instance Name)
vac/crio/we-l1-01 (unique device name).
This identifies the **vac** equipment called **crio**. It is located in the test room which has room name **we-l1-01**. Hence unique device name is **vac/crio/we-l1-01**
The naming convention should be agreed by all and documented. Input from the Tango community who have a lot of experience may be useful.

5.2 Security

Tango provides a server package called TangoAccessControl.
http://www.esrf.eu/computing/cs/Tango/Tango_doc/clients_doc/astor_doc/access_control.html
This can be used to control access to servers and devices. This should be installed on production servers after it has been decided who can access which servers.



```
Access to  orion:11000
  All Users
    Allowed Addresses
      160.103.10.*
      160.103.12.3   (pchqps)
      160.103.14.86   (pcspectro)
      160.103.72.21   (w-syrf-1)
      160.103.73.6   (pclabmag)
    Devices
      ● */*/*
  blissadm
  chaize
  dserver
  epaud
  meyer
  opd01
    Allowed Addresses
      160.103.121.*
    Devices
      ● */*/*
      ● fe/d/1

Devices:  ● Read/Write  ● Read Only
```

Figure 12 Tango Access Control

5.3 Other

Other Standard system tools should be applied
E.g. regular backups, Regression Testing of Tango servers, tickets for problems etc… A person responsible for supporting users should also be identified.

6 Conclusion

Tango has been installed at Ego on several Debian linux machines and has been partly installed on Scientific linux. Several Tango servers have been written and tested mainly for the vacuum control system.

All Tango servers may be controlled by the Astor client but other clients are also available. Several GUI clients have been installed such as jdraw and Taurus (on Debian linux) but operators at Virgo must familiarise themselves with how to use the clients. There are also several programming clients which can be used to integrate with Tango servers (python, C++ and java).

The main tasks still pending are:

1. Choose which Tango clients to use.
2. Move current test servers to production servers
3. A generic interface to the Virgo environment should be created (Tango2cm?)