



# **Status of NICOS at MLZ**

# NICOS at MLZ

- Common instrument control solution for whole MLZ
- Since 2013, taken into operation step by step
- Now in user operation at 20 instruments in Garching and Jülich
- 5 more in commissioning, testing, planning

# Recently finished changes

- “Facility” trees: allow grouping custom functionality by facility+instrument instead of just instrument
- Daemon protocol: extensible transports / serialization to accommodate non-Python clients
- Sandboxing of dry runs / simulation mode
- Infrastructure: CI testing in Docker containers
- Integrated keyring library for external credentials

# Current projects

- Main work still instrument migrations
- Update of software stack: Qt 5 and Python 3 (→ extensive testing)
- Reduce configuration on clients: let the daemon export the guiconfig
- More unified plotting and fitting
- More live data displays (Mantid + custom views)

# Future plans

- Dynamic (“web2.0”) HTML monitor
- Integration of ICAT + new proposal system
- Better “watchdog” service with user-accessible configuration dialogs
- Further protocol changes: file system service through daemon

# Future plans II

- More custom GUIs for common tasks at instruments
- Improve install/setup story, provide consistent Debian/RPM package repositories + Mac/Windows installers for NICOS + non-common dependencies
- Better documentation: better separate development + configuration + user-related aspects

# “Unified” Plotting backend(s)

- PyQwt, Qwt (plots, detector live view)
- Matplotlib (html status monitor)
- Gnuplot (elog)
- XmGrace (scan plots)

# “Unified” Plotting backend(s)

- PyQwt, Qwt (plots, detector live view)
- Matplotlib (html status monitor)
- Gnuplot (elog)
- XmGrace (scan plots)
- Different backends for varying tasks



# “Unified” Plotting backend(s)

- Since 2014 basically two graphics backends
  - GR
  - PyQwt (as fallback) – Qwt
- Python bindings for Qwt are **umaintained**
  - Lacking support for python 3 and Qt5
- Dropping support for Qwt in next release 3.1 (Jan18)
  - Uncluttering plotting interface
  - Python 3 and Qt5 support
  - Extensible Live widget

# Unified Plotting backend



- Universal framework for cross-platform visualization applications
- [Science:gr-framework](#) Repository on [build.opensuse.org](#)
  - {yum, dns, zypper, apt-get} install python-gr
- [WIP] Pre-compiled tarballs for various platforms on  
pip install gr
- Common plot backend in [Plots.jl](#)
- [gr-framework.org](#), MIT License, Runtime 0.27, Wrapper 1.0

A screenshot of a 'Build results' table showing successful builds across various Linux distributions and architectures. Each row includes the OS name, architecture, and status.

OS	Architecture	Status
CentOS_6	i586	suceeded: 2
CentOS_6	x86_64	suceeded: 2
CentOS_7	x86_64	suceeded: 2
Debian_7.0	i586	suceeded: 2
Debian_7.0	x86_64	suceeded: 2
Debian_8.0	i586	suceeded: 2
Debian_8.0	x86_64	suceeded: 2
Debian_9.0	i586	suceeded: 2
Debian_9.0	x86_64	suceeded: 2
Fedora_24	i586	suceeded: 2
Fedora_24	x86_64	suceeded: 2
Fedora_25	i586	suceeded: 2
Fedora_25	x86_64	suceeded: 2
Fedora_26	x86_64	suceeded: 2
RHEL_6	i586	suceeded: 2
RHEL_6	x86_64	suceeded: 2
RHEL_7	x86_64	suceeded: 2
ScientificLinux_6	i586	suceeded: 2
ScientificLinux_6	x86_64	suceeded: 2
ScientificLinux_7	x86_64	suceeded: 2
openSUSE_13.1	i586	suceeded: 2
openSUSE_13.1	x86_64	suceeded: 2
openSUSE_13.2	i586	suceeded: 2
openSUSE_13.2	x86_64	suceeded: 2
xUbuntu_12.04	i586	suceeded: 2
xUbuntu_12.04	x86_64	suceeded: 2
xUbuntu_14.04	i586	suceeded: 2
xUbuntu_14.04	x86_64	suceeded: 2
xUbuntu_16.04	i586	suceeded: 2
xUbuntu_16.04	x86_64	suceeded: 2
xUbuntu_16.10	i586	suceeded: 2
xUbuntu_16.10	x86_64	suceeded: 2
xUbuntu_17.04	i586	suceeded: 2
xUbuntu_17.04	x86_64	suceeded: 2