



Principle and practice of instrument control in neutron facilities - the NICOS system

Jens Krüger

Challenges for Instrument Control

- Completely different types of instruments

Challenges for Instrument Control

- Completely different types of instruments
 - Triple axes

Challenges for Instrument Control

- Completely different types of instruments
 - Triple axes
 - Diffractometer

Challenges for Instrument Control

- Completely different types of instruments
 - Triple axes
 - Diffractometer
 - Reflectometer

Challenges for Instrument Control

- Completely different types of instruments
 - Triple axes
 - Diffractometer
 - Reflectometer
 - Small angle scattering

Challenges for Instrument Control

- Completely different types of instruments
 - Triple axes
 - Diffractometer
 - Reflectometer
 - Small angle scattering
 - Tomography

Challenges for Instrument Control

- Completely different types of instruments
 - Triple axes
 - Diffractometer
 - Reflectometer
 - Small angle scattering
 - Tomography
 - ...

Challenges for Instrument Control

- Completely different types of instruments
 - Triple axes
 - Diffractometer
 - Reflectometer
 - Small angle scattering
 - Tomography
 - ...
- Every instrument is unique!

Challenges for Instrument Control

- Completely different types of instruments
 - Triple axes
 - Diffractometer
 - Reflectometer
 - Small angle scattering
 - Tomography
 - ...
- Every instrument is unique!
- Unified (or standardized) instrument control ?

Challenges for Instrument Control

- Completely different types of instruments
 - Triple axes
 - Diffractometer
 - Reflectometer
 - Small angle scattering
 - Tomography
 - ...
- Every instrument is unique!
- Unified (or standardized) instrument control ?

Typical answer: **No way!**

Challenges for Instrument Control

- Completely different types of instruments
 - Triple axes
 - Diffractometer
 - Reflectometer
 - Small angle scattering
 - Tomography
 - ...
- Every instrument is unique!
- Unified (or standardized) instrument control ?

Typical answer: **No way!**

My answer: Possible

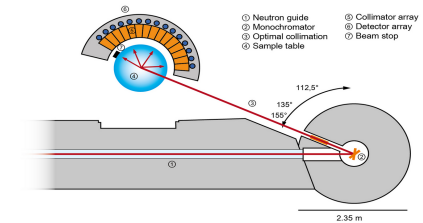
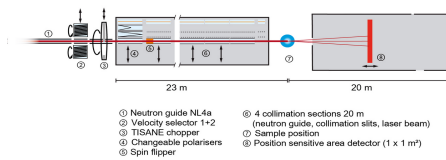
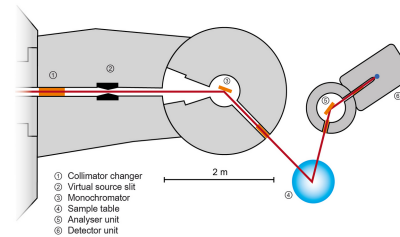
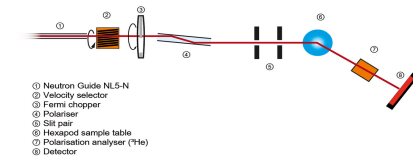
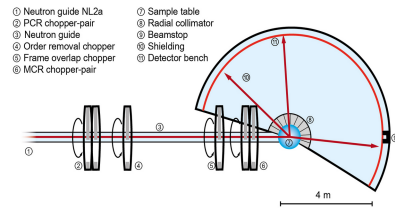
Challenges for Instrument Control

- Completely different types of instruments



Challenges for Instrument Control

- Completely different types of instruments



Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement

Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement
 - Move/Drive a motor/axis

Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement
 - Move/Drive a motor/axis
 - Move/Switch a switch

Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement
 - Move/Drive a motor/axis
 - Move/Switch a switch
 - Move/Drive a voltage/current

Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement
 - Move/Drive a motor/axis
 - Move/Switch a switch
 - Move/Drive a voltage/current
 - Move/Drive a temperature/field (magnetic, electric)

Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement
 - Move/Drive a motor/axis
 - Move/Switch a switch
 - Move/Drive a voltage/current
 - Move/Drive a temperature/field (magnetic, electric)
 - Preset a monitor, counter, detector, ...

Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement
 - Move/Drive a motor/axis
 - Move/Switch a switch
 - Move/Drive a voltage/current
 - Move/Drive a temperature/field (magnetic, electric)
 - Preset a monitor, counter, detector, ...
 - Start/Stop a monitor, counter, detector, ...

Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement
 - Move/Drive a motor/axis
 - Move/Switch a switch
 - Move/Drive a voltage/current
 - Move/Drive a temperature/field (magnetic, electric)
 - Preset a monitor, counter, detector, ...
 - Start/Stop a monitor, counter, detector, ...
 - Wait for finishing of a movement and/or counting

Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement
 - Move/Drive a motor/axis
 - Move/Switch a switch
 - Move/Drive a voltage/current
 - Move/Drive a temperature/field (magnetic, electric)
 - Preset a monitor, counter, detector, ...
 - Start/Stop a monitor, counter, detector, ...
 - Wait for finishing of a movement and/or counting
 - Store signal(s), additional (meta) data (positions, fields, temperature, ...), logs

Challenges for Instrument Control

- Completely different types of instruments
 - Many (in some cases configurable) steps for a single measurement
 - Move/Drive a motor/axis
 - Move/Switch a switch
 - Move/Drive a voltage/current
 - Move/Drive a temperature/field (magnetic, electric)
 - Preset a monitor, counter, detector, ...
 - Start/Stop a monitor, counter, detector, ...
 - Wait for finishing of a movement and/or counting
 - Store signal(s), additional (meta) data (positions, fields, temperature, ...), logs
- Group steps in sequences or more complex devices

Challenges for Instrument Control

- Completely different types of instruments
 - Many (in some cases configurable) steps for a single measurement
 - Move/Drive a motor/axis
 - Move/Switch a switch
 - Move/Drive a voltage/current
 - Move/Drive a temperature/field (magnetic, electric)
 - Preset a monitor, counter, detector, ...
 - Start/Stop a monitor, counter, detector, ...
 - Wait for finishing of a movement and/or counting
 - Store signal(s), additional (meta) data (positions, fields, temperature, ...), logs
- Group steps in sequences or more complex devices

Higher level actions: Counts, Scans, ...

Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement
- Huge number of devices to handle

Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement
- Huge number of devices to handle
 - Motors: dozens of axes are involved and some have to move „synchronised“

Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement
- Huge number of devices to handle
 - Motors: dozens of axes are involved and some have to move „synchronised“
 - Detectors for neutrons and other signals

Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement
- Huge number of devices to handle
 - Motors: dozens of axes are involved and some have to move „synchronised“
 - Detectors for neutrons and other signals
 - Many additional (complex) devices:

Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement
- Huge number of devices to handle
 - Motors: dozens of axes are involved and some have to move „synchronised“
 - Detectors for neutrons and other signals
 - Many additional (complex) devices:
cryostats, magnets, furnaces, frequency generators, sample changer, ...

Challenges for Instrument Control

- Completely different types of instruments
- Many (in some cases configurable) steps for a single measurement
- Huge number of devices to handle
 - Motors: dozens of axes are involved and some have to move „synchronised“
 - Detectors for neutrons and other signals
 - Many additional (complex) devices:
cryostats, magnets, furnaces, frequency generators, sample changer, ...
- Operation with (external) users!



Requirements to Instrument control

Requirements to Instrument control

- Highly automated
 - Goal: achieve a smooth operation

Requirements to Instrument control

- Highly automated
 - Goal: achieve a smooth operation
- All hardware components have to be monitored
 - Goal: reproduce the state

Requirements to Instrument control

- Highly automated
 - Goal: achieve a smooth operation
- All hardware components have to be monitored
 - Goal: reproduce the state
- External users must be able to operate the system within a short time
 - Goal: Effective use of measurement time

Requirements to Instrument control

- Highly automated
 - Goal: achieve a smooth operation
- All hardware components have to be monitored
 - Goal: reproduce the state
- External users must be able to operate the system within a short time
 - Goal: Effective use of measurement time
- Integrate new devices very quickly
 - Goal: Short reaction time to new requests

Requirements to Instrument control

- Highly automated
 - Goal: achieve a smooth operation
- All hardware components have to be monitored
 - Goal: reproduce the state
- External users must be able to operate the system within a short time
 - Goal: Effective use of measurement time
- Integrate new devices very quickly
 - Goal: Short reaction time to new requests
- Common, but flexible
 - Goal: Users need an interface with ability to program

Instrument Control solutions

- Different solutions on market at the Neutron facilities

Instrument Control solutions

- Different solutions on market at the Neutron facilities
- One of them is



NICOS solutions

- Python is implementation and user language

NICOS solutions

- Python is implementation and user language
- NICOS knows
 - Devices (implemented as Python objects)
 - Commands (for routine tasks with full freedom of use)

NICOS solutions

- Python is implementation and user language
- NICOS knows
 - Devices (implemented as Python objects)
 - Commands (for routine tasks with full freedom of use)
- Reconfiguration at runtime with different setups

NICOS solutions

- Python is implementation and user language
- NICOS knows
 - Devices (implemented as Python objects)
 - Commands (for routine tasks with full freedom of use)
- Reconfiguration at runtime with different setups
- Full „dry run“ mode for error checking and timing
 - Syntax checks
 - Limit checks
 - Runtime estimation

NICOS solutions

- Python is implementation and user language
- NICOS knows
 - Devices (implemented as Python objects)
 - Commands (for routine tasks with full freedom of use)
- Reconfiguration at runtime with different setups
- Full „dry run“ mode for error checking and timing
 - Syntax checks
 - Limit checks
 - Runtime estimation
- State collection and automatic archiving

NICOS solutions

- Python is implementation and user language
- NICOS knows
 - Devices (implemented as Python objects)
 - Commands (for routine tasks with full freedom of use)
- Reconfiguration at runtime with different setups
- Full „dry run“ mode for error checking and timing
 - Syntax checks
 - Limit checks
 - Runtime estimation
- State collection and automatic archiving
- Different user interfaces: configurable GUI, console

NICOS solutions

- Python is implementation and user language
- NICOS knows
 - Devices (implemented as Python objects)
 - Commands (for routine tasks with full freedom of use)
- Reconfiguration at runtime with different setups
- Full „dry run“ mode for error checking and timing
 - Syntax checks
 - Limit checks
 - Runtime estimation
- State collection and automatic archiving
- Different user interfaces: configurable GUI, console
- Hardware „independent“ (TANGO, EPICS, SECoP, CARESS, ...)

History of NICOS

- NICOS (Networked Instrument Control System) started in 2002
- Ran on 5 instruments
- Very simple, not a client/server system
- Diverged lines of development and code states (CVS, not code review)

History of NICOS

- NICOS (Networked Instrument Control System) started in 2002
- Ran on 5 instruments
- Very simple, not a client/server system
- Diverged lines of development and code states (CVS, not code review)
- 2009 discussions about modernizing between user and instrument control groups, start rewriting of NICOS (client/server and GUI), first commit 2009/11/10

History of NICOS

- NICOS (Networked Instrument Control System) started in 2002
- Ran on 5 instruments
- Very simple, not a client/server system
- Diverged lines of development and code states (CVS, not code review)
- 2009 discussions about modernizing between user and instrument control groups, start rewriting of NICOS (client/server and GUI), first commit 2009/11/10
- 2011 switch to git/gerrit/Jenkins/Redmine in the development

History of NICOS

- NICOS (Networked Instrument Control System) started in 2002
- Ran on 5 instruments
- Very simple, not a client/server system
- Diverged lines of development and code states (CVS, not code review)
- 2009 discussions about modernizing between user and instrument control groups, start rewriting of NICOS (client/server and GUI), first commit 2009/11/10
- 2011 switch to git/gerrit/Jenkins/Redmine in the development
- 2012 „NICOS 2“ was stable and fully featured

History of NICOS

- NICOS (Networked Instrument Control System) started in 2002
- Ran on 5 instruments
- Very simple, not a client/server system
- Diverged lines of development and code states (CVS, not code review)
- 2009 discussions about modernizing between user and instrument control groups, start rewriting of NICOS (client/server and GUI), first commit 2009/11/10
- 2011 switch to git/gerrit/Jenkins/Redmine in the development
- 2012 „NICOS 2“ was stable and fully featured
- 2013 MLZ decision, NICOS is standard control system

History of NICOS

- NICOS (Networked Instrument Control System) started in 2002
- Ran on 5 instruments
- Very simple, not a client/server system
- Diverged lines of development and code states (CVS, not code review)
- 2009 discussions about modernizing between user and instrument control groups, start rewriting of NICOS (client/server and GUI), first commit 2009/11/10
- 2011 switch to git/gerrit/Jenkins/Redmine in the development
- 2012 „NICOS 2“ was stable and fully featured
- 2013 MLZ decision, NICOS is standard control system
- 2015 collaboration with ESS started

History of NICOS

- NICOS (Networked Instrument Control System) started in 2002
- Ran on 5 instruments
- Very simple, not a client/server system
- Diverged lines of development and code states (CVS, not code review)
- 2009 discussions about modernizing between user and instrument control groups, start rewriting of NICOS (client/server and GUI), first commit 2009/11/10
- 2011 switch to git/gerrit/Jenkins/Redmine in the development
- 2012 „NICOS 2“ was stable and fully featured
- 2013 MLZ decision, NICOS is standard control system
- 2015 collaboration with ESS started
- 2018 PSI joined the NICOS developer group

History of NICOS

- NICOS (Networked Instrument Control System) started in 2002
- Ran on 5 instruments
- Very simple, not a client/server system
- Diverged lines of development and code states (CVS, not code review)
- 2009 discussions about modernizing between user and instrument control groups, start rewriting of NICOS (client/server and GUI), first commit 2009/11/10
- 2011 switch to git/gerrit/Jenkins/Redmine in the development
- 2012 „NICOS 2“ was stable and fully featured
- 2013 MLZ decision, NICOS is standard control system
- 2015 collaboration with ESS started
- 2018 PSI joined the NICOS developer group
- 2018 INL (US, Aaron Craft) joined the NICOS users group

History of NICOS

- NICOS (Networked Instrument Control System) started in 2002
- Ran on 5 instruments
- Very simple, not a client/server system
- Diverged lines of development and code states (CVS, not code review)
- 2009 discussions about modernizing between user and instrument control groups, start rewriting of NICOS (client/server and GUI), first commit 2009/11/10
- 2011 switch to git/gerrit/Jenkins/Redmine in the development
- 2012 „NICOS 2“ was stable and fully featured
- 2013 MLZ decision, NICOS is standard control system
- 2015 collaboration with ESS started
- 2018 PSI joined the NICOS developer group
- 2018 INL (US, Aaron Craft) joined the NICOS users group
- 2019/2020 MGML (Charles University Prague), LAHN (Argentina) facility, and TU Vienna started to use NICOS

History of NICOS

- NICOS (Networked Instrument Control System) started in 2002
- Ran on 5 instruments
- Very simple, not a client/server system
- Diverged lines of development and code states (CVS, not code review)
- 2009 discussions about modernizing between user and instrument control groups, start rewriting of NICOS (client/server and GUI), first commit 2009/11/10
- 2011 switch to git/gerrit/Jenkins/Redmine in the development
- 2012 „NICOS 2“ was stable and fully featured
- 2013 MLZ decision, NICOS is standard control system
- 2015 collaboration with ESS started
- 2018 PSI joined the NICOS developer group
- 2018 INL (US, Aaron Craft) joined the NICOS users group
- 2019/2020 MGML (Charles University Prague), LAHN (Argentina) facility, and TU Vienna started to use NICOS
- 2025 BNC ?

Design principles of NICOS

Design principles of NICOS

- Accessibility:
 - Powerful to „power users“
 - Easy to learn for new users

Design principles of NICOS

- Accessibility:
 - Powerful to „power users“
 - Easy to learn for new users
- Flexibility:
 - Integration of devices by instrument scientists
 - Quick reaction to new experiment conditions

Design principles of NICOS

- Accessibility:
 - Powerful to „power users“
 - Easy to learn for new users
- Flexibility:
 - Integration of devices by instrument scientists
 - Quick reaction to new experiment conditions
- Reliability:
 - Permanent storing of experiment state
 - Recovering previous state

Design principles of NICOS

- Accessibility:
 - Powerful to „power users“
 - Easy to learn for new users
- Flexibility:
 - Integration of devices by instrument scientists
 - Quick reaction to new experiment conditions
- Reliability:
 - Permanent storing of experiment state
 - Recovering previous state
- Lightweight:
 - Less code is better
 - Reuse existing code as much as possible
 - Better to configure than to code

Design principles of NICOS

- Network based:
 - Multi client
 - Multi user

Design principles of NICOS

- Network based:
 - Multi client
 - Multi user
- Science driven
 - Close cooperation with instrument scientists and/or users
 - No feature without any need

Design principles of NICOS

- Network based:
 - Multi client
 - Multi user
- Science driven
 - Close cooperation with instrument scientists and/or users
 - No feature without any need
- Open source
 - Everybody is welcome to use and contribute

Supported instrument types

- Triple axes

Supported instrument types

- Triple axes
- Reflectometer (Neutrons and X-Ray)

Supported instrument types

- Triple axes
- Reflectometer (Neutrons and X-Ray)
- Tomography

Supported instrument types

- Triple axes
- Reflectometer (Neutrons and X-Ray)
- Tomography
- Powder diffractometer

Supported instrument types

- Triple axes
- Reflectometer (Neutrons and X-Ray)
- Tomography
- Powder diffractometer
- Residual stress (Neutron and X-Ray)

Supported instrument types

- Triple axes
- Reflectometer (Neutrons and X-Ray)
- Tomography
- Powder diffractometer
- Residual stress (Neutron and X-Ray)
- Single crystal spectrometer

Supported instrument types

- Triple axes
- Reflectometer (Neutrons and X-Ray)
- Tomography
- Powder diffractometer
- Residual stress (Neutron and X-Ray)
- Single crystal spectrometer
- Small angle scattering

Supported instrument types

- Triple axes
- Reflectometer (Neutrons and X-Ray)
- Tomography
- Powder diffractometer
- Residual stress (Neutron and X-Ray)
- Single crystal spectrometer
- Small angle scattering
- ...

Setups

- Configuration of an instrument is divided into several „setups“

Setups

- Configuration of an instrument is divided into several „setups“
- A setup is a simple Python file defining devices and their parameters

```
description = 'MIRA1 monochromator'
group = 'lowlevel'

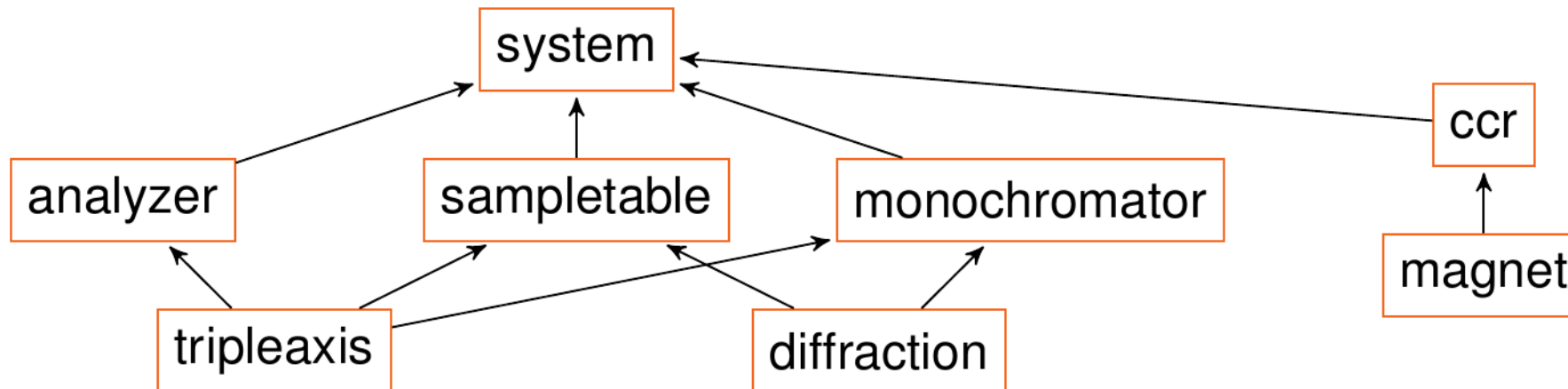
includes = ['base', 'sample', 'alias_mono']

tango_base = 'tango://miractrl.mira.frm2.tum.de:10000/mira/'

devices = dict(
    co_mltt = device('nicos.devices.entangle.Sensor',
        visibility = (),
        tangodevice = tango_base + 'monol/mtt_enc',
        unit = 'deg',
        precision = 0.05,
    ),
    mo_mltt = device('nicos.devices.entangle.Motor',
        tangodevice = tango_base + 'monol/mtt_mot',
        visibility = (),
        precision = 0.05,
    ),
    mltt = device('nicos_mlz.mira.devices.axis.HoveringAxis',
        description = 'monochromator two-theta angle',
        abslimits = (-50.0, 0),
        motor = 'mo_mltt',
        coder = 'co_mltt',
        startdelay = 1,
        stopdelay = 4,
        switch = 'air_mono',
        switchvalues = (0, 1),
        fmtstr = '%.3f',
        precision = 0.05,
    ),
    ...
```

Setups

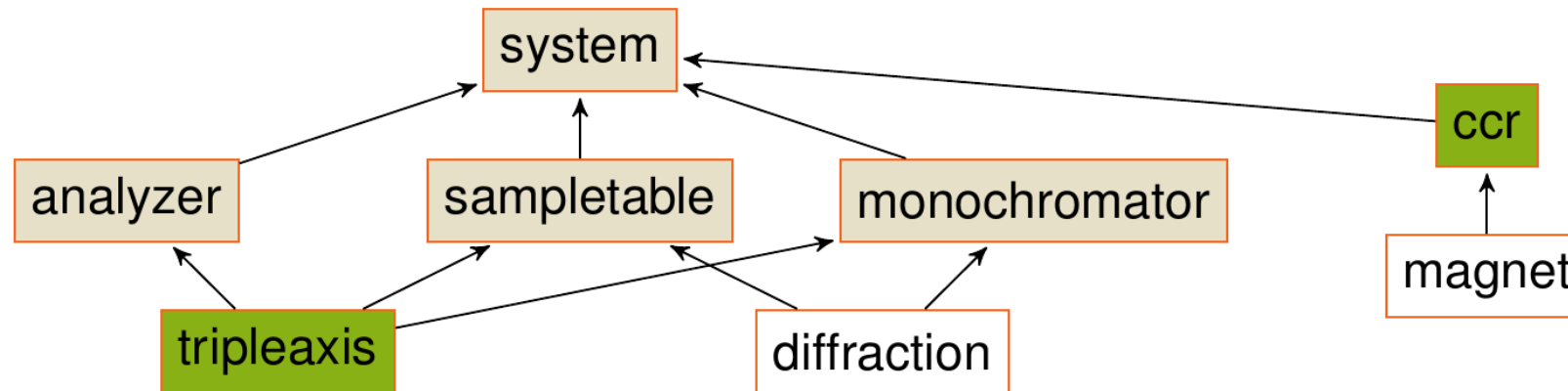
- Configuration of an instrument is divided into several „setups“
- A setup is a simple Python file defining devices and their parameters
- A setup may depend on other setups (used or use)



Setups

- Configuration of an instrument is divided into several „setups“
- A setup is a simple Python file defining devices and their parameters
- A setup may depend on other setups (used or use)

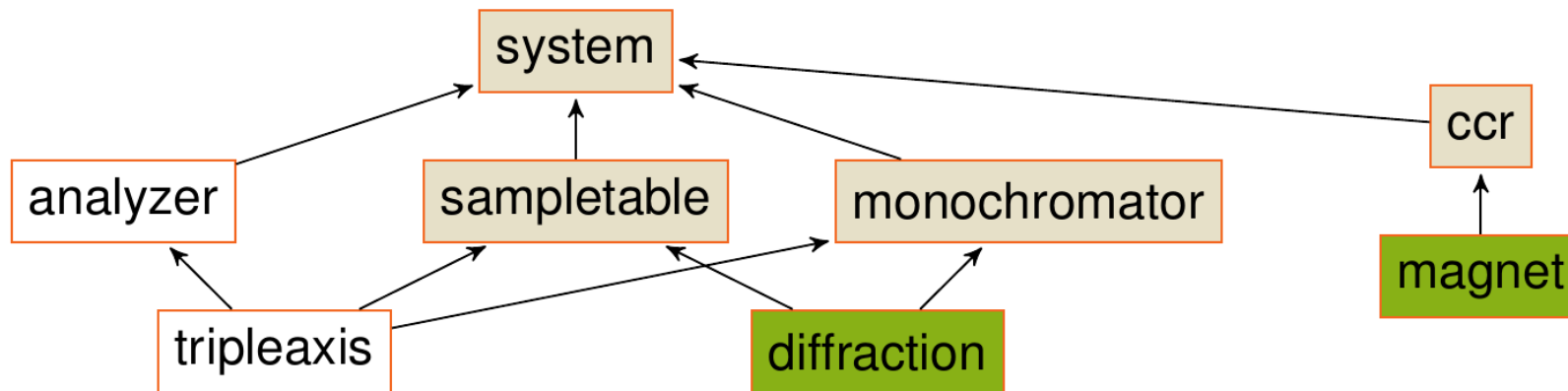
```
NewSetup('tripleaxis', 'ccr')
```



Setups

- Configuration of an instrument is divided into several „setups“
- A setup is a simple Python file defining devices and their parameters
- A setup may depend on other setups (used or use)

```
NewSetup('diffraction', 'magnet')
```



Devices in NICOS

Devices in NICOS

- NICOS encapsulates access to hardware in „Devices“

Devices in NICOS

- NICOS encapsulates access to hardware in „Devices“
- Each device will be represented by a configurable Python class

Devices in NICOS

- NICOS encapsulates access to hardware in „Devices“
- Each device will be represented by a configurable Python class
- NICOS has a hierarchy of devices with a set of base classes
 - define the interface, which has to be implemented in derived classes

Devices in NICOS

- NICOS encapsulates access to hardware in „Devices“
- Each device will be represented by a configurable Python class
- NICOS has a hierarchy of devices with a set of base classes
 - define the interface, which has to be implemented in derived classes
- A number of generic device and helper classes

Devices in NICOS

- NICOS encapsulates access to hardware in „Devices“
- Each device will be represented by a configurable Python class
- NICOS has a hierarchy of devices with a set of base classes
 - define the interface, which has to be implemented in derived classes
- A number of generic device and helper classes
- Device classes may inherit other classes or simply use them

Devices in NICOS

- NICOS encapsulates access to hardware in „Devices“
- Each device will be represented by a configurable Python class
- NICOS has a hierarchy of devices with a set of base classes
 - define the interface, which has to be implemented in derived classes
- A number of generic device and helper classes
- Device classes may inherit other classes or simply use them
- Configuration parameters:
 - Device parameters (read/write access during runtime)
 - Values and their changes will be archived

Devices in NICOS

- NICOS encapsulates access to hardware in „Devices“
- Each device will be represented by a configurable Python class
- NICOS has a hierarchy of devices with a set of base classes
 - define the interface, which has to be implemented in derived classes
- A number of generic device and helper classes
- Device classes may inherit other classes or simply use them
- Configuration parameters:
 - Device parameters (read/write access during runtime)
 - Values and their changes will be archived
- Device value and status will be automatically archived

User interface

User interface

- Most basic user interface: Python **command** prompt

User interface

- Most basic user interface: Python **command** prompt
- Simple commands:
 - `read(device)`, `move(device, target)`, `status(device)`,...

User interface

- Most basic user interface: Python **command** prompt
- Simple commands:
 - `read(device), move(device, target), status(device),...`
- Scanning commands:
 - `scan(device, start, step, num, preset), ...`

User interface

- Most basic user interface: Python **command** prompt
- Simple commands:
 - `read(device), move(device, target), status(device),...`
- Scanning commands:
 - `scan(device, start, step, num, preset), ...`
- Integrated help system:
 - `help(object), ListDevices(), ListSetups(),...`

User interface

- Most basic user interface: Python **command** prompt
- Simple commands:
 - `read(device), move(device, target), status(device),...`
- Scanning commands:
 - `scan(device, start, step, num, preset), ...`
- Integrated help system:
 - `help(object), ListDevices(), ListSetups(),...`
- Python control structures
 - `loops, conditions, variables, ...`

User interface

- Most basic user interface: Python **command** prompt
- Simple commands:
 - `read(device)`, `move(device, target)`, `status(device)`,...
- Scanning commands:
 - `scan(device, start, step, num, preset)`, ...
- Integrated help system:
 - `help(object)`, `ListDevices()`, `ListSetups()`,...
- Python control structures
 - `loops`, `conditions`, `variables`, ...
- Execution environment is safe against user errors
 - **no overriding of devices and commands**

Graphical user interface

- Client-Server architecture: more than one connection possible

Graphical user interface

- Client-Server architecture: more than one connection possible
- Command line window

Graphical user interface

The screenshot displays the NICOS graphical user interface (GUI) for a neutron scattering experiment. The main window is titled "NICOS - guest at localhost:1301".

Command Line: The command line shows the execution of the `qscan` command with parameters: `qscan((1, 0, 0, 0), (0.002, 0, 0, 0), 10, t=1, kf=1.4)`. The command is highlighted in blue.

Select command: Below the command line, there are buttons for "Device", "Scan", and "Other".

Status Bar: The status bar indicates: "Status: Scan H :: Moving to start :: Waiting :: estimated 1 second left / kf -> 1.400, ana -> 1.400, att -> -83.957".

Data Table: A table of scan data is displayed, showing parameters like scan number, time, and various detector readings. The table is titled "Scan H" and contains multiple rows of data.

Experiment Information and Setup: A panel on the right side of the GUI provides details about the experiment setup, including the proposal name, title, users, local contact, and various experimental parameters like detectors, scans, and remark.

NICOS devices: A panel on the right side of the GUI lists the available devices and their current status. The devices are organized into a tree structure, with "cryo" and "source" being the primary categories.

Name	Value	Status
cryo		
• ana	2.001 K	
• Ts	2.001 K	
source		
• ReactorPower	19.845 MW	idle
system		
• demo		
• Exp		
lastpoint	0	
lastscan	0	
• Sample	Gd3CdB7	
• Space	194.867 GiB	194.87 GiB free
• UBahn	7, 17, 27, 37, 47 min	
tas		
• alpha	0.000 deg	idle
• ana	1.409 A-1	theta=idle, twotheta=virtual moving
• ath	-41.978 deg	idle
• att	-83.518 deg	virtual moving
• ec	None	chi=idle, omega=idle
• echi	0.000 deg	idle
• Ef	4.116 meV	theta=idle, twotheta=virtual moving
• Ei	4.978 meV	theta=idle, twotheta=idle
• ephi	0.000 deg	idle
• kf	1.409 A-1	theta=idle, twotheta=virtual moving
• ki	1.550 A-1	theta=idle, twotheta=idle
• Lad	400.000 mm	
• Lms	1000.000 mm	

Graphical user interface

The screenshot displays the NICOS graphical user interface (GUI) for a neutron scattering experiment. The window title is "NICOS - guest at localhost:1301".

Command Line: The command line shows the command `qscan((1, 0, 0, 0), (0.002, 0, 0, 0), 10, t=1, kf=1.4)` entered. The "Run" button is highlighted with a red circle.

Experiment Information and Setup: The right sidebar contains the following information:

- Proposal: p0
- Title: NICOS demo experiment
- Users: Local Contact H. Maier-Leibnitz <heinz.maier-leibnitz@frm2.tum.de>
- Setups: tas
- Samples: Gd3CdB7
- Environments: Detectors vdet, Scans T, Ts
- Remark: (empty)
- NICOS devices: Enter search expression

Status: The status bar indicates "Scan H :: Moving to start :: Waiting :: estimated 1 second left / kf -> 1.400, ana -> 1.400, att -> -83.957".

Data Table: The main area displays a table of scan data. The table has columns for scan number, time, and various parameters. The data is organized into groups for different scan types (e.g., powder line, M spurion).

Scan #	h	k	l	E	mono	ana	sth	phi	T	Ts	t	mon	ctr
11:29:37	0.9800	-0.0000	0.0000	0.0000	1.400	1.400	308.926	77.852	2.001	2.001	0.000	0	0
11:29:37	0.9820	-0.0000	0.0000	0.0000	1.400	1.400	309.021	78.041	2.001	2.001	0.000	0	0
11:29:37	0.9840	-0.0000	0.0000	0.0000	1.400	1.400	309.115	78.231	2.001	2.001	0.000	0	0
11:29:37	0.9860	-0.0000	0.0000	0.0000	1.400	1.400	309.210	78.420	2.001	2.001	0.000	0	0
11:29:37	0.9880	-0.0000	0.0000	0.0000	1.400	1.400	309.305	78.610	2.001	2.001	0.000	0	0
11:29:37	0.9900	-0.0000	0.0000	0.0000	1.400	1.400	309.400	78.800	2.001	2.001	0.000	0	0
11:29:37	0.9920	-0.0000	0.0000	0.0000	1.400	1.400	309.495	78.990	2.001	2.001	0.000	0	0
11:29:37	0.9940	-0.0000	0.0000	0.0000	1.400	1.400	309.590	79.181	2.001	2.001	0.000	0	0
11:29:37	0.9960	-0.0000	0.0000	0.0000	1.400	1.400	309.686	79.372	2.001	2.001	0.000	0	0
11:29:37	0.9980	-0.0000	0.0000	0.0000	1.400	1.400	309.781	79.563	2.001	2.001	0.000	0	0
11:29:37	1.0000	-0.0000	0.0000	0.0000	1.400	1.400	309.877	79.754	2.001	2.001	0.000	0	0
11:29:37	1.0020	-0.0000	0.0000	0.0000	1.400	1.400	309.973	79.946	2.001	2.001	0.000	0	0
11:29:37	1.0040	-0.0000	0.0000	0.0000	1.400	1.400	310.069	80.137	2.001	2.001	0.000	0	0
11:29:37	1.0060	-0.0000	0.0000	0.0000	1.400	1.400	310.165	80.330	2.001	2.001	0.000	0	0
11:29:37	1.0080	-0.0000	0.0000	0.0000	1.400	1.400	310.261	80.522	2.001	2.001	0.000	0	0
11:29:37	1.0100	-0.0000	0.0000	0.0000	1.400	1.400	310.357	80.715	2.001	2.001	0.000	0	0
11:29:37	1.0120	-0.0000	0.0000	0.0000	1.400	1.400	310.454	80.908	2.001	2.001	0.000	0	0
11:29:37	1.0140	-0.0000	0.0000	0.0000	1.400	1.400	310.550	81.101	2.001	2.001	0.000	0	0
11:29:37	1.0160	-0.0000	0.0000	0.0000	1.400	1.400	310.647	81.294	2.001	2.001	0.000	0	0
11:29:37	1.0180	-0.0000	0.0000	0.0000	1.400	1.400	310.744	81.488	2.001	2.001	0.000	0	0
11:29:37	1.0200	-0.0000	0.0000	0.0000	1.400	1.400	310.841	81.682	2.001	2.001	0.000	0	0

Parameters Table: The bottom right section shows a table of parameters and their values.

Name	Value	Status
cryo		
ana	2.001 K	
Ts	2.001 K	
source		
ReactorPower	19.845 MW	idle
system		
demo		
Exp		
lastpoint	0	
lastscan	0	
Sample	Gd3CdB7	
Space	194.867 GiB	194.87 GiB free
UBahn	7, 17, 27, 37, 47 min	
tas		
alpha	0.000 deg	idle
ana	1.409 A-1	theta=idle, twotheta=virtual moving
ath	-41.978 deg	idle
att	-83.518 deg	virtual moving
ec	None	chi=idle, omega=idle
echi	0.000 deg	idle
Ef	4.116 meV	theta=idle, twotheta=virtual moving
Ei	4.978 meV	theta=idle, twotheta=idle
ephi	0.000 deg	idle
kf	1.409 A-1	theta=idle, twotheta=virtual moving
ki	1.550 A-1	theta=idle, twotheta=idle
Lad	400.000 mm	
Lms	1000.000 mm	

Graphical user interface

The screenshot displays the NICOS graphical user interface (GUI) running on a local host. The interface is divided into several sections:

- Command line:** A text area where the command `qscan((1, 0, 0, 0), (0.002, 0, 0, 0), 10, t=1, kf=1.4)` is entered. Below it, a "Select command:" dropdown menu shows "Dry run" and "Run" buttons. A red circle highlights the "Run" button.
- Experiment Information and Setup:** A panel on the right side containing fields for "Proposal" (p0), "Title" (NICOS demo experiment), "Users", "Local Contact" (H. Maier-Leibnitz <heinz.maier-leibnitz@frm2.tum.de>), "Setups" (tas), "Samples" (Gd3CdB7), "Environments", "Detectors vdet", "Scans T, Ts", and "Remark".
- NICOS devices:** A section for entering search expressions.
- Data Table:** A large table displaying scan data. The table has columns for "Scan number", "h", "k", "l", "E meV", "mono", "ana", "sth", "phi", "T", "Ts", "t", "mon", and "ctr". The data shows a series of scans with various parameters and values.

The status bar at the bottom indicates: "Status: Scan H :: Moving to start :: Waiting :: estimated 1 second left / kf -> 1.400, ana -> 1.400, att -> -83.957".

Graphical user interface

The screenshot displays the NICOS graphical user interface (GUI) for a neutron scattering experiment. The main window is titled "NICOS - guest at localhost:1301".

Command Line: The command line shows the command `qscan((1, 0, 0, 0), (0.002, 0, 0, 0), 10, t=1, kf=1.4)` entered. The "Run" button is highlighted with a red circle.

Experiment Information and Setup: The right sidebar contains the following information:

- Proposal: p0
- Title: NICOS demo experiment
- Users: Local Contact H. Maier-Leibnitz <heinz.maier-leibnitz@frm2.tum.de>
- Setups: tas
- Samples: Gd3CdB7
- Environments: Detectors vdet, Scans T, Ts
- Remark: (empty)
- NICOS devices: Enter search expression

Data Table: The main area displays a table of scan data. The status bar indicates "Status: Scan H :: Moving to start :: Waiting :: estimated 1 second left / kf -> 1.400, ana -> 1.400, att -> -83.957".

(sim) scan number:	h	k	l	E	mono	ana	sth	phi	T	Ts	t	mon	ctr
(sim) 1/21	0.9800	-0.0000	0.0000	0.0000	1.400	1.400	309.926	77.852	2.001	2.001	0.000	0	0
(sim) 2/21	0.9820	-0.0000	0.0000	0.0000	1.400	1.400	309.021	78.041	2.001	2.001	0.000	0	0
(sim) 3/21	0.9840	-0.0000	0.0000	0.0000	1.400	1.400	309.115	78.231	2.001	2.001	0.000	0	0
(sim) 4/21	0.9860	-0.0000	0.0000	0.0000	1.400	1.400	309.210	78.420	2.001	2.001	0.000	0	0
(sim) 5/21	0.9880	-0.0000	0.0000	0.0000	1.400	1.400	309.305	78.610	2.001	2.001	0.000	0	0
(sim) 6/21	0.9900	-0.0000	0.0000	0.0000	1.400	1.400	309.400	78.800	2.001	2.001	0.000	0	0
(sim) 7/21	0.9920	-0.0000	0.0000	0.0000	1.400	1.400	309.495	78.990	2.001	2.001	0.000	0	0
(sim) 8/21	0.9940	-0.0000	0.0000	0.0000	1.400	1.400	309.590	79.181	2.001	2.001	0.000	0	0
(sim) 9/21	0.9960	-0.0000	0.0000	0.0000	1.400	1.400	309.686	79.372	2.001	2.001	0.000	0	0
(sim) 10/21	0.9980	-0.0000	0.0000	0.0000	1.400	1.400	309.781	79.563	2.001	2.001	0.000	0	0
(sim) 11/21	1.0000	-0.0000	0.0000	0.0000	1.400	1.400	309.877	79.754	2.001	2.001	0.000	0	0
(sim) 12/21	1.0020	-0.0000	0.0000	0.0000	1.400	1.400	309.973	79.946	2.001	2.001	0.000	0	0
(sim) 13/21	1.0040	-0.0000	0.0000	0.0000	1.400	1.400	310.069	80.137	2.001	2.001	0.000	0	0
(sim) 14/21	1.0060	-0.0000	0.0000	0.0000	1.400	1.400	310.165	80.330	2.001	2.001	0.000	0	0
(sim) 15/21	1.0080	-0.0000	0.0000	0.0000	1.400	1.400	310.261	80.522	2.001	2.001	0.000	0	0
(sim) 16/21	1.0100	-0.0000	0.0000	0.0000	1.400	1.400	310.357	80.715	2.001	2.001	0.000	0	0
(sim) 17/21	1.0120	-0.0000	0.0000	0.0000	1.400	1.400	310.454	80.908	2.001	2.001	0.000	0	0
(sim) 18/21	1.0140	-0.0000	0.0000	0.0000	1.400	1.400	310.550	81.101	2.001	2.001	0.000	0	0
(sim) 19/21	1.0160	-0.0000	0.0000	0.0000	1.400	1.400	310.647	81.294	2.001	2.001	0.000	0	0
(sim) 20/21	1.0180	-0.0000	0.0000	0.0000	1.400	1.400	310.744	81.488	2.001	2.001	0.000	0	0
(sim) 21/21	1.0200	-0.0000	0.0000	0.0000	1.400	1.400	310.841	81.682	2.001	2.001	0.000	0	0

The table shows the progression of the scan, with parameters like h, k, l, E, mono, ana, sth, phi, T, Ts, t, mon, and ctr. The status bar indicates the scan is finished at 2021-07-16 11:29:37.

Graphical user interface

The screenshot displays the NICOS graphical user interface (GUI) running on a local host. The interface is divided into several sections:

- Command line:** A text area where the command `qscan((1, 0, 0, 0), (0.002, 0, 0, 0), 10, t=1, kf=1.4)` is entered. Below it are buttons for `Dry run` and `Run`, which are circled in red.
- Experiment Information and Setup:** A panel on the right containing fields for Proposal (p0), Title (NICOS demo experiment), Users, Local Contact (H. Maier-Leibnitz <heinz.maier-leibnitz@frm2.tum.de>), Setups (tas), Samples (Gd3CdB7), Environments, Detectors (vdet), Scans (T, Ts), and a Remark field.
- NICOS devices:** A search bar and a table listing various devices and their status.
- Status:** A section showing the current scan status: `Scan H :: Moving to start :: Waiting :: estimated 1 second left / kf -> 1.400, ana -> 1.400, att -> -83.957`.
- Data Table:** A large table with columns for scan number, time, and various parameters. The table contains multiple rows of data, including scan numbers, times, and values for parameters like `h`, `k`, `l`, `E`, `mono`, `ana`, `sth`, `phi`, `T`, `Ts`, `t`, `mon`, and `ctr`.

The table data is as follows:

Scan number	Time	h	k	l	E	mono	ana	sth	phi	T	Ts	t	mon	ctr
11:29:37	(sim)	0.9800	-0.0000	0.0000	0.0000	1.400	1.400	308.926	77.852	2.001	2.001	0.000	0	0
11:29:37	(sim) 1/21	0.9800	-0.0000	0.0000	0.0000	1.400	1.400	308.926	77.852	2.001	2.001	0.000	0	0
11:29:37	(sim) 2/21	0.9820	-0.0000	0.0000	0.0000	1.400	1.400	309.021	78.041	2.001	2.001	0.000	0	0
11:29:37	(sim) 3/21	0.9840	-0.0000	0.0000	0.0000	1.400	1.400	309.115	78.231	2.001	2.001	0.000	0	0
11:29:37	(sim) 4/21	0.9860	-0.0000	0.0000	0.0000	1.400	1.400	309.210	78.420	2.001	2.001	0.000	0	0
11:29:37	(sim) 5/21	0.9880	-0.0000	0.0000	0.0000	1.400	1.400	309.305	78.610	2.001	2.001	0.000	0	0
11:29:37	(sim) 6/21	0.9900	-0.0000	0.0000	0.0000	1.400	1.400	309.400	78.800	2.001	2.001	0.000	0	0
11:29:37	(sim) 7/21	0.9920	-0.0000	0.0000	0.0000	1.400	1.400	309.495	78.990	2.001	2.001	0.000	0	0
11:29:37	(sim) 8/21	0.9940	-0.0000	0.0000	0.0000	1.400	1.400	309.590	79.181	2.001	2.001	0.000	0	0
11:29:37	(sim) 9/21	0.9960	-0.0000	0.0000	0.0000	1.400	1.400	309.686	79.372	2.001	2.001	0.000	0	0
11:29:37	(sim) 10/21	0.9980	-0.0000	0.0000	0.0000	1.400	1.400	309.781	79.563	2.001	2.001	0.000	0	0
11:29:37	(sim) 11/21	1.0000	-0.0000	0.0000	0.0000	1.400	1.400	309.877	79.754	2.001	2.001	0.000	0	0
11:29:37	(sim) 12/21	1.0020	-0.0000	0.0000	0.0000	1.400	1.400	309.973	79.946	2.001	2.001	0.000	0	0
11:29:37	(sim) 13/21	1.0040	-0.0000	0.0000	0.0000	1.400	1.400	310.069	80.137	2.001	2.001	0.000	0	0
11:29:37	(sim) 14/21	1.0060	-0.0000	0.0000	0.0000	1.400	1.400	310.165	80.330	2.001	2.001	0.000	0	0
11:29:37	(sim) 15/21	1.0080	-0.0000	0.0000	0.0000	1.400	1.400	310.261	80.522	2.001	2.001	0.000	0	0
11:29:37	(sim) 16/21	1.0100	-0.0000	0.0000	0.0000	1.400	1.400	310.357	80.715	2.001	2.001	0.000	0	0
11:29:37	(sim) 17/21	1.0120	-0.0000	0.0000	0.0000	1.400	1.400	310.454	80.908	2.001	2.001	0.000	0	0
11:29:37	(sim) 18/21	1.0140	-0.0000	0.0000	0.0000	1.400	1.400	310.550	81.101	2.001	2.001	0.000	0	0
11:29:37	(sim) 19/21	1.0160	-0.0000	0.0000	0.0000	1.400	1.400	310.647	81.294	2.001	2.001	0.000	0	0
11:29:37	(sim) 20/21	1.0180	-0.0000	0.0000	0.0000	1.400	1.400	310.744	81.488	2.001	2.001	0.000	0	0
11:29:37	(sim) 21/21	1.0200	-0.0000	0.0000	0.0000	1.400	1.400	310.841	81.682	2.001	2.001	0.000	0	0

The table also includes a section for the `tas` setup, listing parameters like `alpha`, `ana`, `ath`, `att`, `ec`, `echi`, `ef`, `ei`, `ephi`, `kf`, `ki`, `lad`, `lms`, and `...` with their respective values and status.

Graphical user interface

The screenshot displays the NICOS graphical user interface (GUI) for a neutron scattering experiment. The main window is titled "NICOS - guest at localhost:1301".

Command Line: The command line shows the command `qscan((1, 0, 0, 0), (0.002, 0, 0, 0), 10, t=1, kf=1.4)` entered. The "Run" button is highlighted with a red circle.

Experiment Information and Setup: The right sidebar contains the following information:

- Proposal: p0
- Title: NICOS demo experiment
- Users: Local Contact H. Maier-Leibnitz <heinz.maier-leibnitz@frm2.tum.de>
- Setups: tas
- Samples: Gd3CdB7
- Environments: Detectors vdet, Scans T, Ts
- Remark: (empty)

Status: The status bar indicates "Scan H :: Moving to start :: Waiting :: estimated 1 second left / kf -> 1.400, ana -> 1.400, att -> -83.957".

Data Table: The main area displays a table of scan data. The table has columns for scan number, time, and various parameters. The data is organized into groups for different scans (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22).

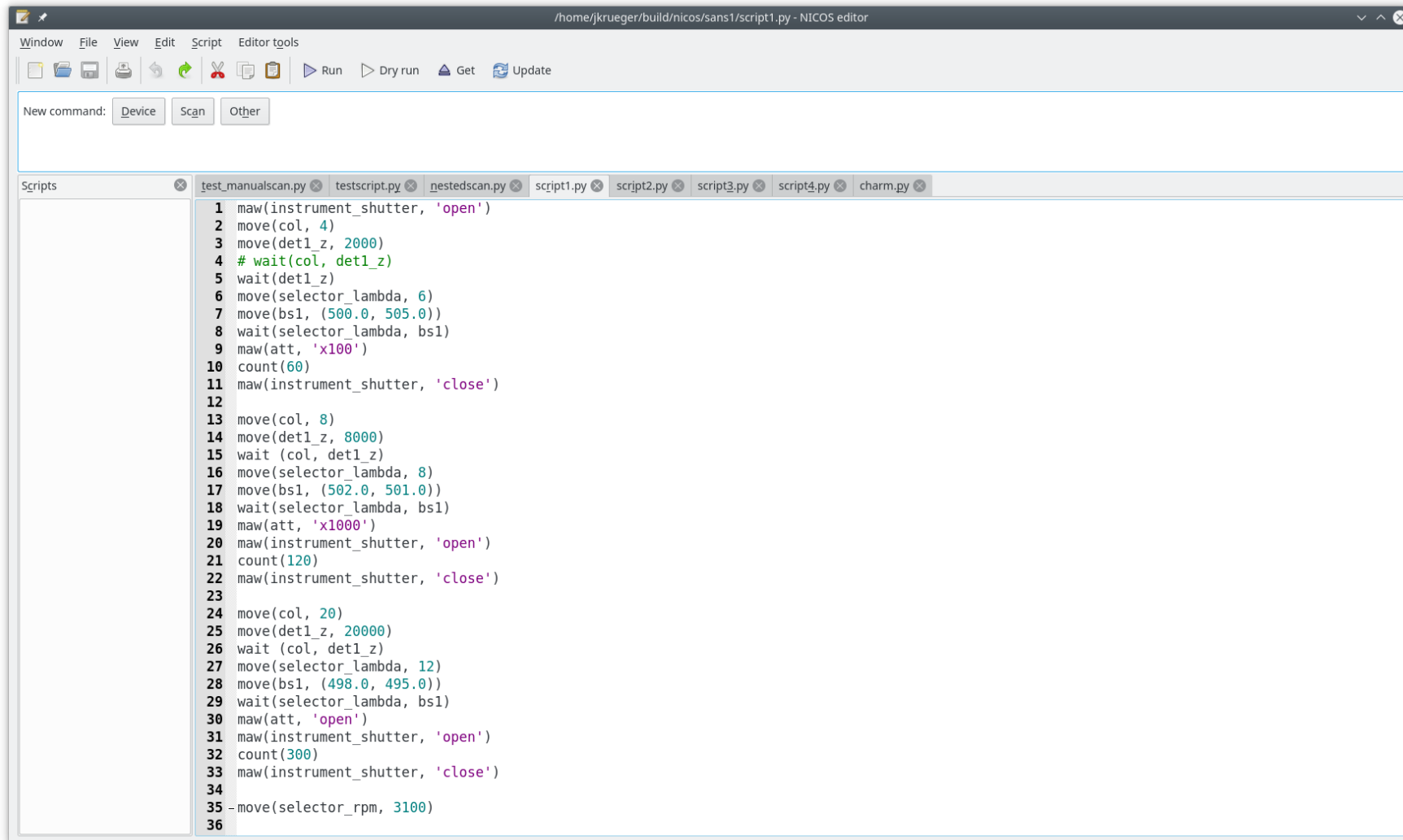
Scan	Time	h	k	l	E	mono	ana	sth	phi	T	Ts	t	mon	ctr
1	11:29:37	0.9800	-0.0000	0.0000	0.0000	1.400	1.400	308.926	77.852	2.001	2.001	0.000	0	0
2	11:29:37	0.9820	-0.0000	0.0000	0.0000	1.400	1.400	309.021	78.041	2.001	2.001	0.000	0	0
3	11:29:37	0.9840	-0.0000	0.0000	0.0000	1.400	1.400	309.115	78.231	2.001	2.001	0.000	0	0
4	11:29:37	0.9860	-0.0000	0.0000	0.0000	1.400	1.400	309.210	78.420	2.001	2.001	0.000	0	0
5	11:29:37	0.9880	-0.0000	0.0000	0.0000	1.400	1.400	309.305	78.610	2.001	2.001	0.000	0	0
6	11:29:37	0.9900	-0.0000	0.0000	0.0000	1.400	1.400	309.400	78.800	2.001	2.001	0.000	0	0
7	11:29:37	0.9920	-0.0000	0.0000	0.0000	1.400	1.400	309.495	78.990	2.001	2.001	0.000	0	0
8	11:29:37	0.9940	-0.0000	0.0000	0.0000	1.400	1.400	309.590	79.181	2.001	2.001	0.000	0	0
9	11:29:37	0.9960	-0.0000	0.0000	0.0000	1.400	1.400	309.686	79.372	2.001	2.001	0.000	0	0
10	11:29:37	0.9980	-0.0000	0.0000	0.0000	1.400	1.400	309.781	79.563	2.001	2.001	0.000	0	0
11	11:29:37	1.0000	-0.0000	0.0000	0.0000	1.400	1.400	309.877	79.754	2.001	2.001	0.000	0	0
12	11:29:37	1.0020	-0.0000	0.0000	0.0000	1.400	1.400	309.973	79.946	2.001	2.001	0.000	0	0
13	11:29:37	1.0040	-0.0000	0.0000	0.0000	1.400	1.400	310.069	80.137	2.001	2.001	0.000	0	0
14	11:29:37	1.0060	-0.0000	0.0000	0.0000	1.400	1.400	310.165	80.330	2.001	2.001	0.000	0	0
15	11:29:37	1.0080	-0.0000	0.0000	0.0000	1.400	1.400	310.261	80.522	2.001	2.001	0.000	0	0
16	11:29:37	1.0100	-0.0000	0.0000	0.0000	1.400	1.400	310.357	80.715	2.001	2.001	0.000	0	0
17	11:29:37	1.0120	-0.0000	0.0000	0.0000	1.400	1.400	310.454	80.908	2.001	2.001	0.000	0	0
18	11:29:37	1.0140	-0.0000	0.0000	0.0000	1.400	1.400	310.550	81.101	2.001	2.001	0.000	0	0
19	11:29:37	1.0160	-0.0000	0.0000	0.0000	1.400	1.400	310.647	81.294	2.001	2.001	0.000	0	0
20	11:29:37	1.0180	-0.0000	0.0000	0.0000	1.400	1.400	310.744	81.488	2.001	2.001	0.000	0	0
21	11:29:37	1.0200	-0.0000	0.0000	0.0000	1.400	1.400	310.841	81.682	2.001	2.001	0.000	0	0

Parameters: The bottom right section shows the parameters for the experiment, including "alpha", "ana", "ath", "att", "ec", "echi", "Ef", "Ei", "ephi", "kf", "ki", "Lad", "Lms", and "t".

Graphical user interface

- Client-Server architecture: more than one connection possible
- Command line window
- User script editor

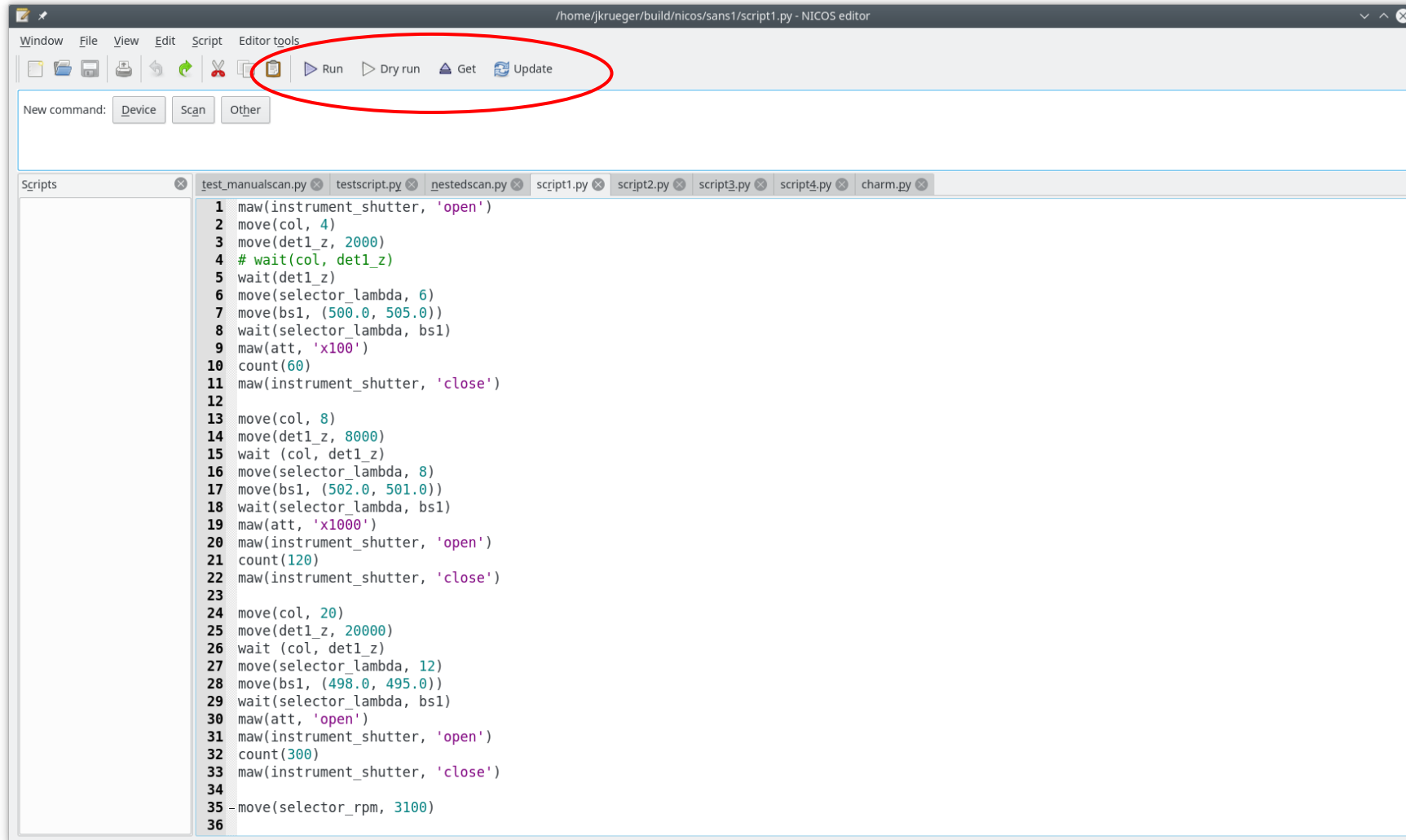
Graphical user interface



The screenshot displays the Nicos editor window, titled "/home/jkrueger/build/nicos/sans1/script1.py - NICOS editor". The interface includes a menu bar (Window, File, View, Edit, Script, Editor tools) and a toolbar with icons for file operations, execution (Run, Dry run), and updates (Get, Update). Below the toolbar is a "New command:" section with buttons for "Device", "Scan", and "Other". The main area shows a list of open scripts on the left and a large text editor on the right. The active script, "script1.py", contains a Python script for controlling an instrument, with lines numbered 1 to 36. The script includes commands for moving components, waiting, and setting parameters.

```
1 maw(instrument_shutter, 'open')
2 move(col, 4)
3 move(det1_z, 2000)
4 # wait(col, det1_z)
5 wait(det1_z)
6 move(selector_lambda, 6)
7 move(bs1, (500.0, 505.0))
8 wait(selector_lambda, bs1)
9 maw(att, 'x100')
10 count(60)
11 maw(instrument_shutter, 'close')
12
13 move(col, 8)
14 move(det1_z, 8000)
15 wait (col, det1_z)
16 move(selector_lambda, 8)
17 move(bs1, (502.0, 501.0))
18 wait(selector_lambda, bs1)
19 maw(att, 'x1000')
20 maw(instrument_shutter, 'open')
21 count(120)
22 maw(instrument_shutter, 'close')
23
24 move(col, 20)
25 move(det1_z, 20000)
26 wait (col, det1_z)
27 move(selector_lambda, 12)
28 move(bs1, (498.0, 495.0))
29 wait(selector_lambda, bs1)
30 maw(att, 'open')
31 maw(instrument_shutter, 'open')
32 count(300)
33 maw(instrument_shutter, 'close')
34
35 -move(selector_rpm, 3100)
36
```

Graphical user interface

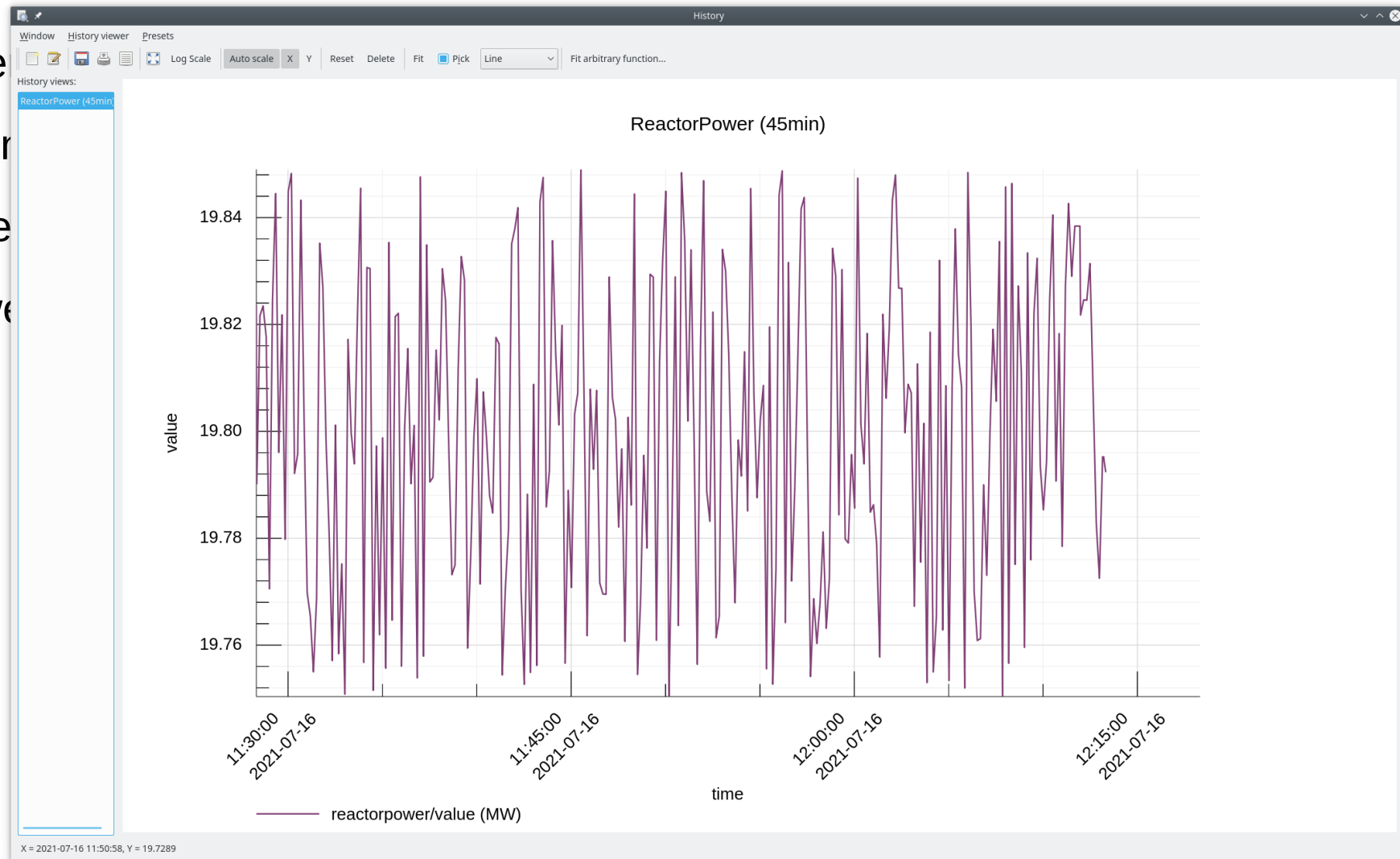


Graphical user interface

- Client-Server architecture: more than one connection possible
- Command line window
- User script editor
- History viewer: Display of archived data

Graphical user interface

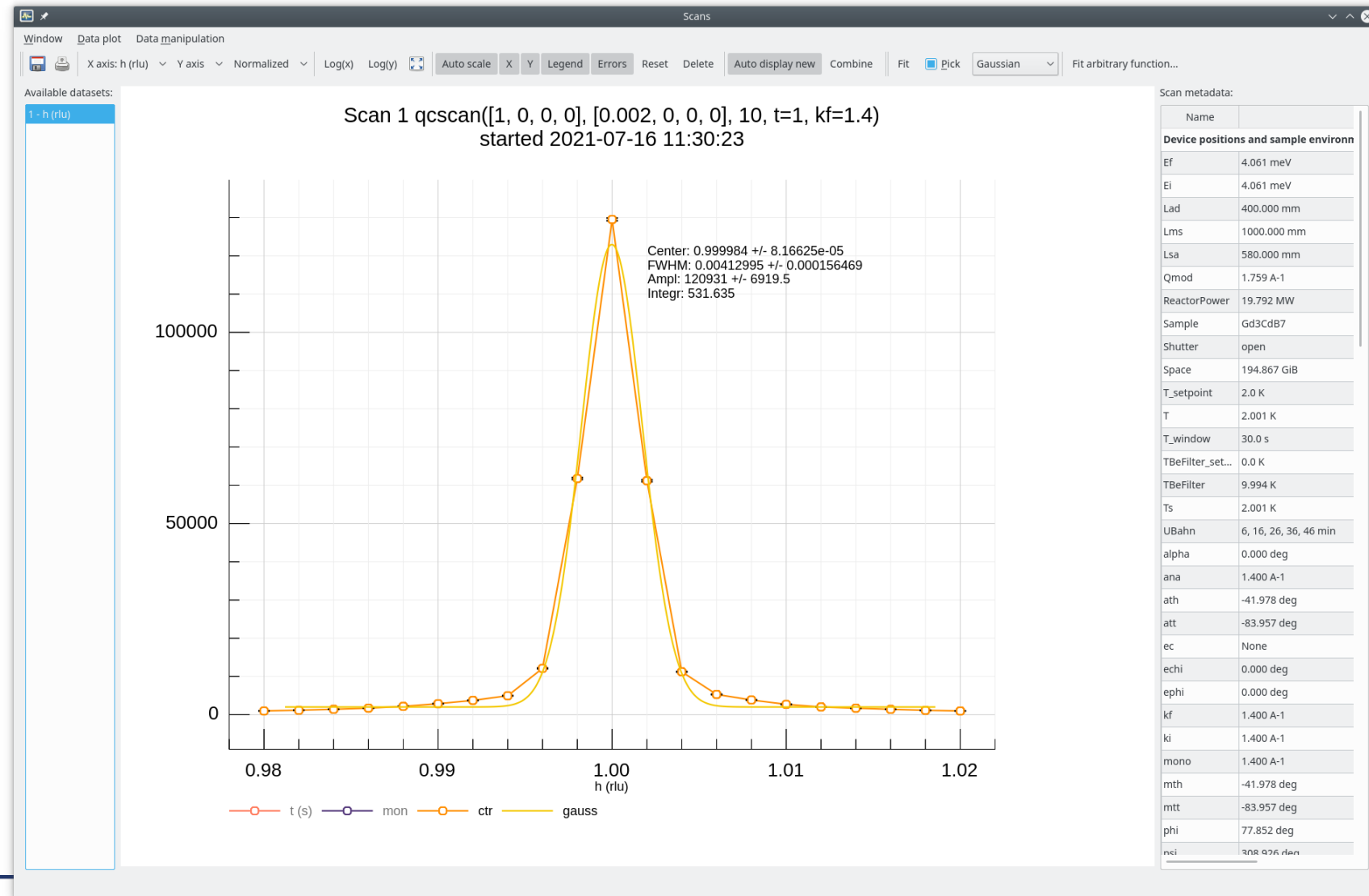
- Client-Server
- Command line
- User script editor
- History viewer



Graphical user interface

- Client-Server architecture: more than one connection possible
- Command line window
- User script editor
- History viewer: Display of archived data
- On-line data display with preliminary fitting

Graphical user interface



Graphical user interface



Graphical user interface

- Client-Server architecture: more than one connection possible
- Command line window
- User script editor
- History viewer: Display of archived data
- On-line data display with preliminary fitting
- Electronic logbook to show the experiment history

Graphical user interface

The screenshot displays the MLZ Logbook graphical user interface. The interface is divided into several sections:

- Window / Browser / Logbook:** The top navigation bar.
- Messages:** A list of messages on the right side, including "New setup: tas", "New standard environment: T, Ts", and "New standard detectors: vdet".
- Contents:** A sidebar on the left with a tree view showing the hierarchy of the logbook, including "Experiment p0: NICOS demo experiment", "New sample: Gd3CdB7", "New standard environment: T, Ts", "New standard detectors: vdet", and "New standard environment: T, Ts".
- Logbook:** The main area displaying the log of events. It includes a "Script finished: setting up NICOS" message and a detailed scan log for a qscan experiment.

The scan log for the qscan experiment is as follows:

```
>>> [guest 2021-07-16 11:29:44] qscan((1, 0, 0, 0), (0.002, 0, 0, 0), 10, t=1, kf=1.4)
tas : scan mode is now CKF at 1.400 A-1
WARNING: creating new empty file counter file at data/counters
Starting scan: qscan((1, 0, 0, 0), (0.002, 0, 0, 0), 10, t=1, kf=1.4)
Started at: 2021-07-16 11:30:23
Scan number: 1
Filename: p0_00000001.dat
```

#	h rlu	k rlu	l rlu	E meV	mono A-1	ana A-1	sth deg	phi deg	T K	Ts K	t s	mon cts	ctr cts
1/21	0.9800	-0.0000	0.0000	0.0000	1.400	1.400	308.926	77.852	2.001	2.001	1.000	35715	994
2/21	0.9820	-0.0000	0.0000	0.0000	1.400	1.400	309.021	78.041	2.001	2.001	1.000	35732	1177
3/21	0.9840	-0.0000	0.0000	0.0000	1.400	1.400	309.115	78.231	2.001	2.001	1.000	35635	1424
4/21	0.9860	-0.0000	0.0000	0.0000	1.400	1.400	309.210	78.420	2.001	2.001	1.000	35414	1729
5/21	0.9880	-0.0000	0.0000	0.0000	1.400	1.400	309.305	78.610	2.001	2.001	1.000	35917	2195
6/21	0.9900	-0.0000	0.0000	0.0000	1.400	1.400	309.400	78.800	2.001	2.001	1.000	35488	2884
7/21	0.9920	-0.0000	0.0000	0.0000	1.400	1.400	309.495	78.990	2.001	2.001	1.000	35924	3749
8/21	0.9940	-0.0000	0.0000	0.0000	1.400	1.400	309.590	79.181	2.001	2.001	1.000	35782	4962
9/21	0.9960	-0.0000	0.0000	0.0000	1.400	1.400	309.686	79.372	2.001	2.001	1.000	35777	12121
10/21	0.9980	-0.0000	0.0000	0.0000	1.400	1.400	309.781	79.563	2.001	2.001	1.000	35660	61765
11/21	1.0000	-0.0000	0.0000	0.0000	1.400	1.400	309.877	79.754	2.001	2.001	1.000	35984	129515
12/21	1.0020	-0.0000	0.0000	0.0000	1.400	1.400	309.973	79.946	2.001	2.001	1.000	35820	61190
13/21	1.0040	-0.0000	0.0000	0.0000	1.400	1.400	310.069	80.137	2.001	2.001	1.000	35924	11232
14/21	1.0060	-0.0000	0.0000	0.0000	1.400	1.400	310.165	80.330	2.001	2.001	1.000	35579	5310
15/21	1.0080	-0.0000	0.0000	0.0000	1.400	1.400	310.261	80.522	2.001	2.001	1.000	35344	3861
16/21	1.0100	-0.0000	0.0000	0.0000	1.400	1.400	310.357	80.715	2.001	2.001	1.000	35843	2711
17/21	1.0120	-0.0000	0.0000	0.0000	1.400	1.400	310.454	80.908	2.001	2.001	1.000	35604	2041
18/21	1.0140	-0.0000	0.0000	0.0000	1.400	1.400	310.550	81.101	2.001	2.001	1.000	35484	1716
19/21	1.0160	-0.0000	0.0000	0.0000	1.400	1.400	310.647	81.294	2.001	2.001	1.000	35588	1417
20/21	1.0180	-0.0000	0.0000	0.0000	1.400	1.400	310.744	81.488	2.001	2.001	1.000	35721	1152
21/21	1.0200	-0.0000	0.0000	0.0000	1.400	1.400	310.841	81.682	2.001	2.001	1.000	36142	986

Finished at: 2021-07-16 11:30:53

Scan#	Points	h (rlu)	k (rlu)	l (rlu)	E (meV)	mono (A-1)	ana (A-1)	sth (deg)	phi (deg)	T (K)	Ts (K)	Plot	Data
1	21	0.9800 - 1.0200	-0.0000 - 0.0000	0.0000	0.0000	1.400	1.400	308.926 - 310.841	77.852 - 81.682	2.001	2.001	Lin / Log	File

Graphical user interface

- Client-Server architecture: more than one connection possible
- Command line window
- User script editor
- History viewer: Display of archived data
- On-line data display with preliminary fitting
- Electronic logbook to show the experiment history
- Configurable and dynamic status monitor

Graphical user interface

- Client-Server architecture: more than one connection possible
- Command line window
- User script editor
- History viewer: Display of archived data
- On-line data display with preliminary fitting
- Electronic logbook to show the experiment history
- Configurable and dynamic status monitor
- Configurable and dynamic

NICOS - guest at localhost:1301

Application Script control Output Live data Windows Tools Help

Connect Exit View Editor Scans History Live data Logbook Log files Errors Watchdog

Experiment Information and Setup

Proposal: p0

Title: NICOS demo experiment

Users:

Local Contact: H. Maier-Leibnitz <heinz.maier-leibnitz@frm2.tum.de>

Setups: startup

Samples: Gd3CdB7

Environments:

Detectors:

Scans:

Remark:

Command line

>>

Select command: Device Scan Other

```
1 | NicosSetup()
```

```
[16:46:22] >>> [system 2025-01-27 16:46:22] setting up NICOS
[16:46:22] loading setup 'system' (system setup)
[16:46:22] importing module 'nicos.commands.standard'...
[16:46:22] loading setup 'startup' (NICOS demo startup setup)
[16:46:22] Cache      : now connected to localhost:14869
[16:46:22] creating device 'Exp' (experiment object)...
[16:46:22] creating device 'Sample' (sample object)...
[16:46:22] creating device 'LogSpace' (Space on log drive)...
[16:46:22] creating device 'Space' (The amount of free space for storing data)...
[16:46:23] creating device 'conssink'...
[16:46:23] creating device 'demo' (demo instrument)...
[16:46:23] creating device 'dmnsink'...
[16:46:23] creating device 'filesink'...
[16:46:23] =====
[16:46:23] Welcome to the NICOS demo.
[16:46:23] Run one of the following commands to set up either a triple-axis
[16:46:23] or a SANS demo setup:
[16:46:23]   > NewSetup('tas')
[16:46:23]   > NewSetup('sans')
[16:46:23] setups loaded: startup
[16:46:23] checking master status...
[16:46:23] switched to master mode
```

NICOS devices

Enter search expression

Name	Value	Status
system		
Exp		
lastpoint	0	
lastscan	0	
Sample	Gd3CdB7	
Space	42.831 GiB	42.83 GiB free

NICOS - guest at localhost:1301

Application Output Script control File View Edit Script Tools Help

NICOS Instrument: DEMO Experiment: NICOS demo experiment Connected

Setup Experiment Instrument

Instrument interaction
Batch file generation
Detector Image
History
Logs
Elog
Finish Experiment

New experiment

Enter a proposal number:

Current experiment

Experiment title:

Users:

Local contact:

Sample name:

Script path:

Notifications
(one email address
per line):

Send data
(one email address
per line):

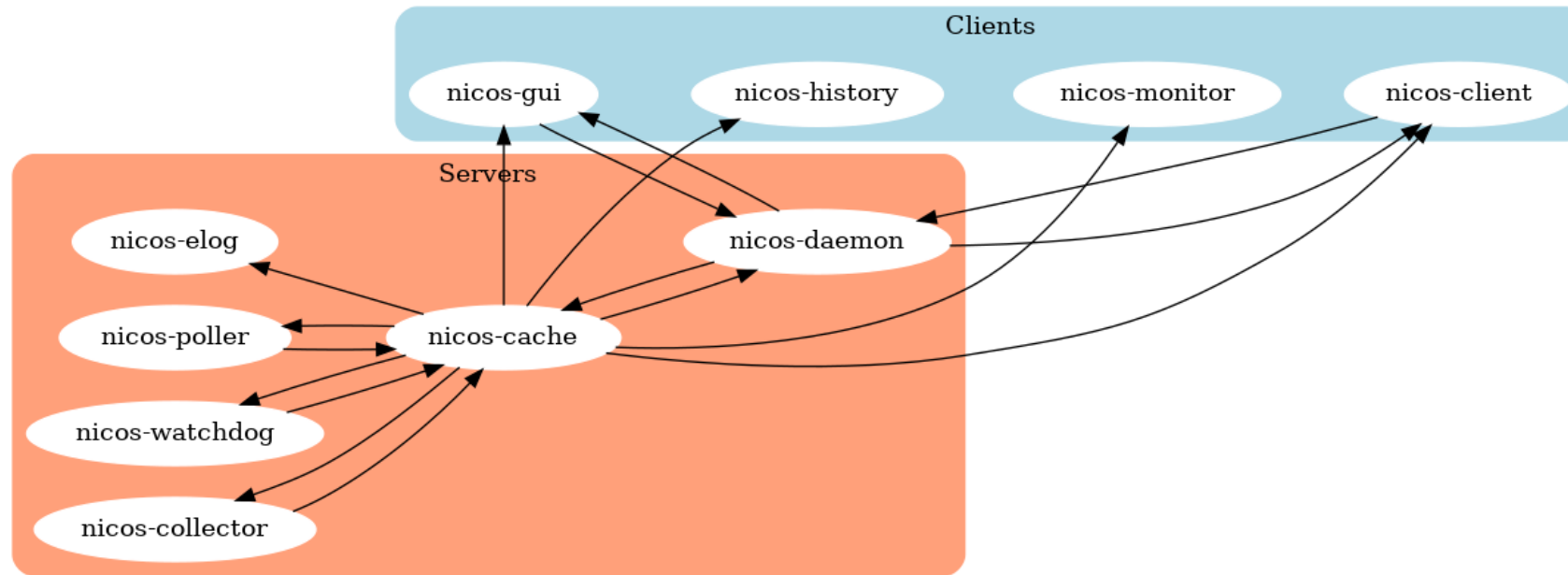
☐ Do not continue scripts after fatal errors

✓ Anwenden

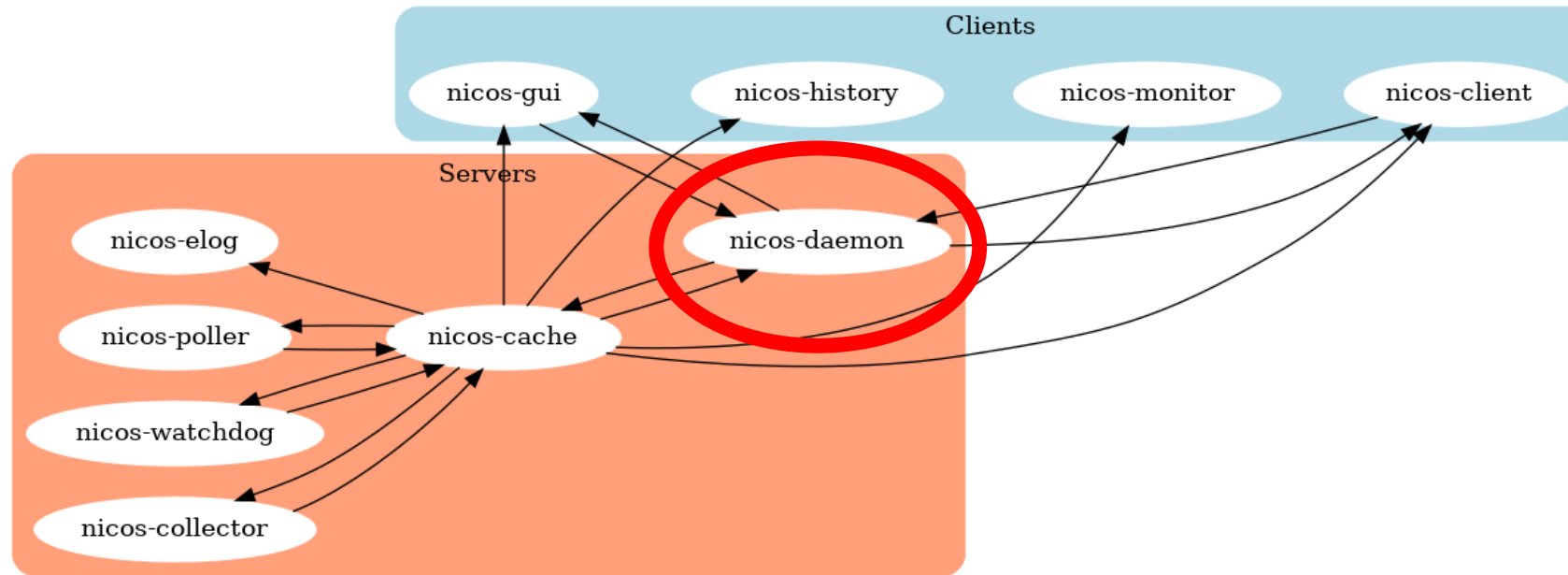
Graphical user interface

- Client-Server architecture: more than one connection possible
- Command line window
- User script editor
- History viewer: Display of archived data
- On-line data display with preliminary fitting
- Electronic logbook to show the experiment history
- Configurable and dynamic status monitor
- Configurable and dynamic
- Extendable by custom panels

Components

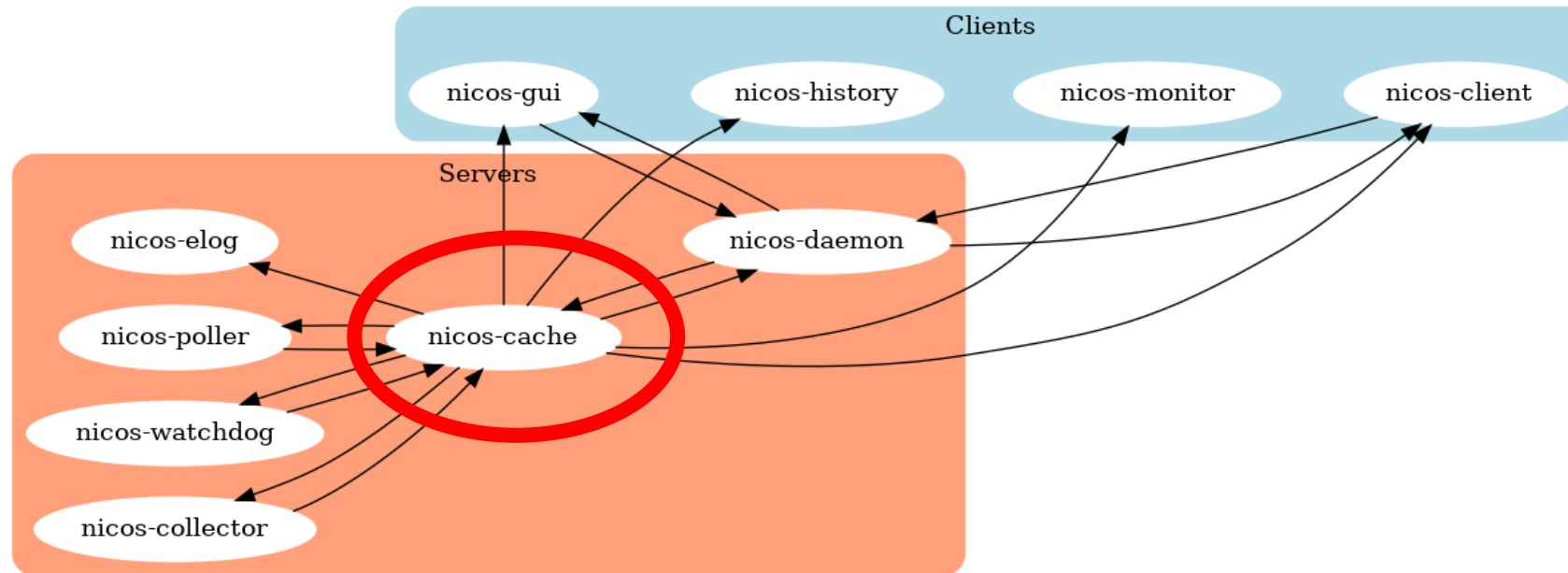


Components



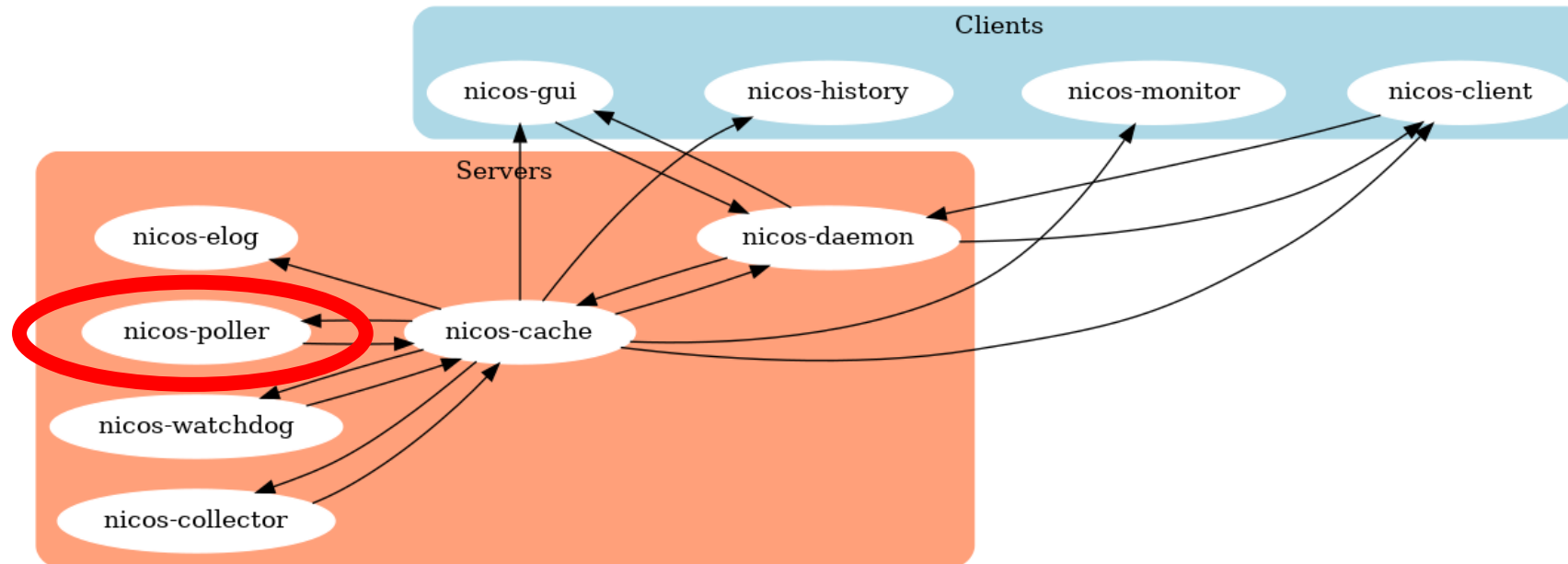
- Execution daemon: executes user input

Components



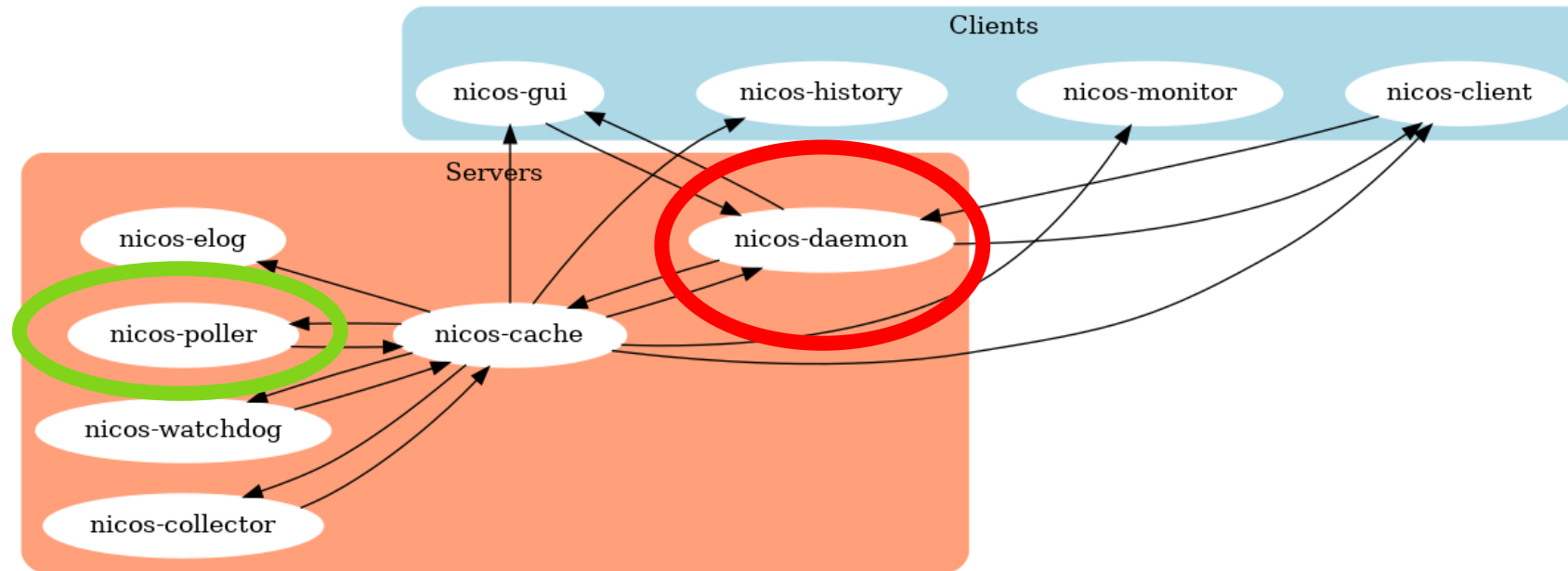
- Cache: save states/values/parameters, pushes value/state changes

Components



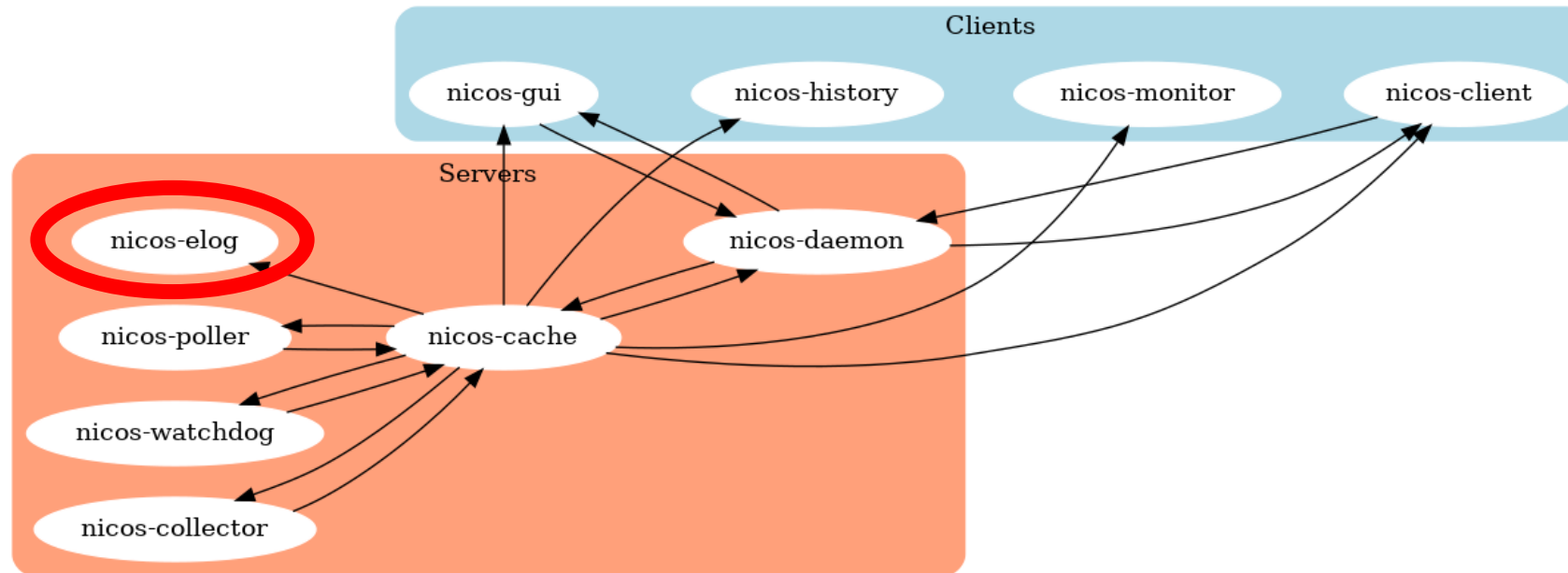
- Poller: fetches periodically values/parameter/states from devices

Components



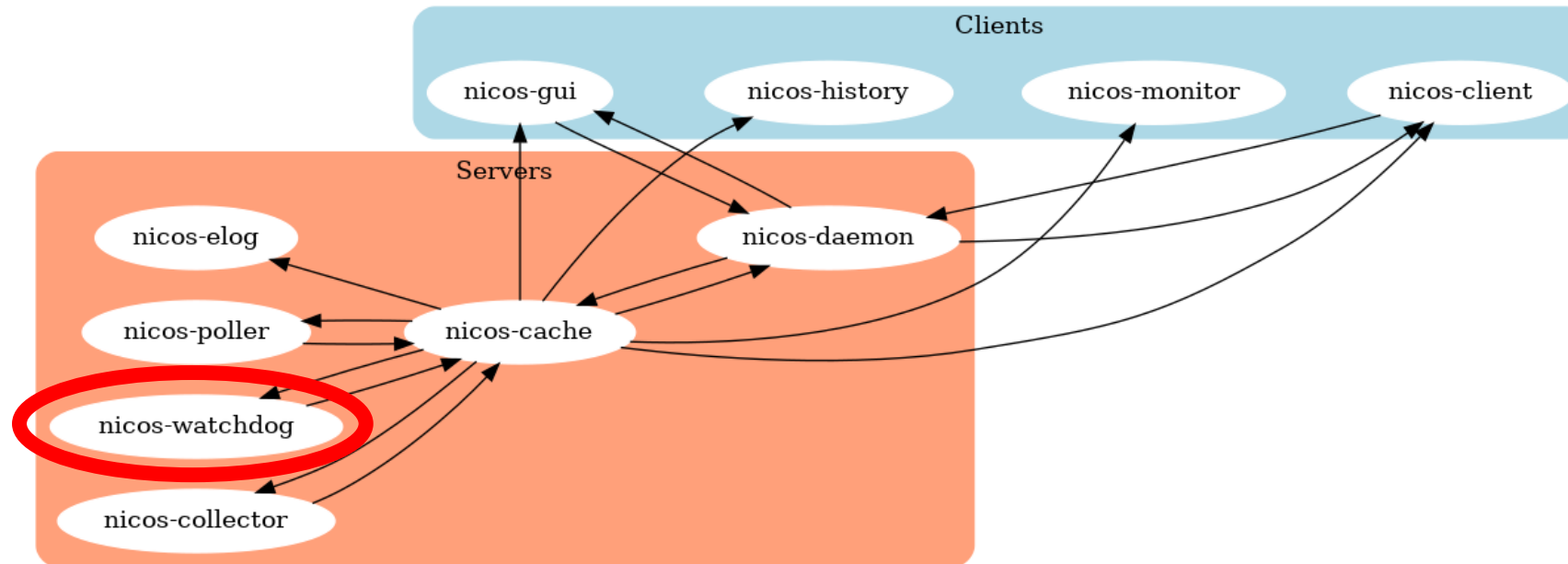
- Daemon and poller (read-only) interact with hardware

Components



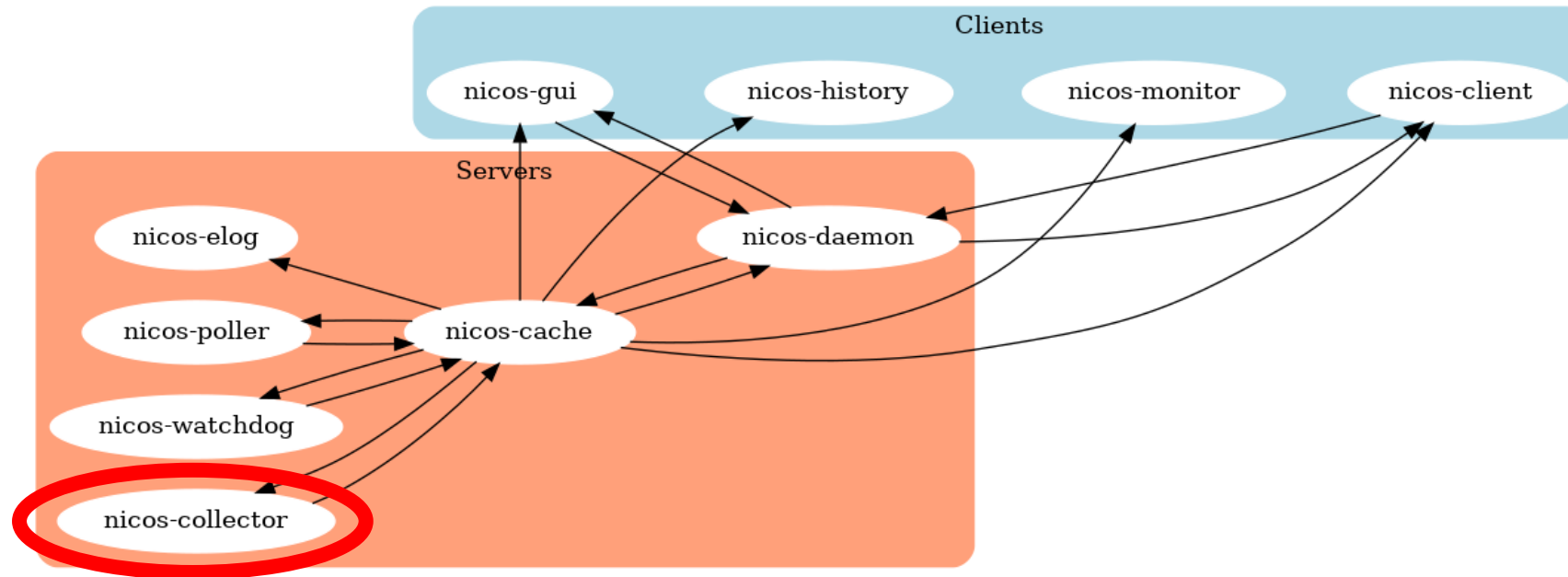
- Electronic logbook: experiment history in user friendly form (ASCII, HTML, external „ELN“, ...)

Components



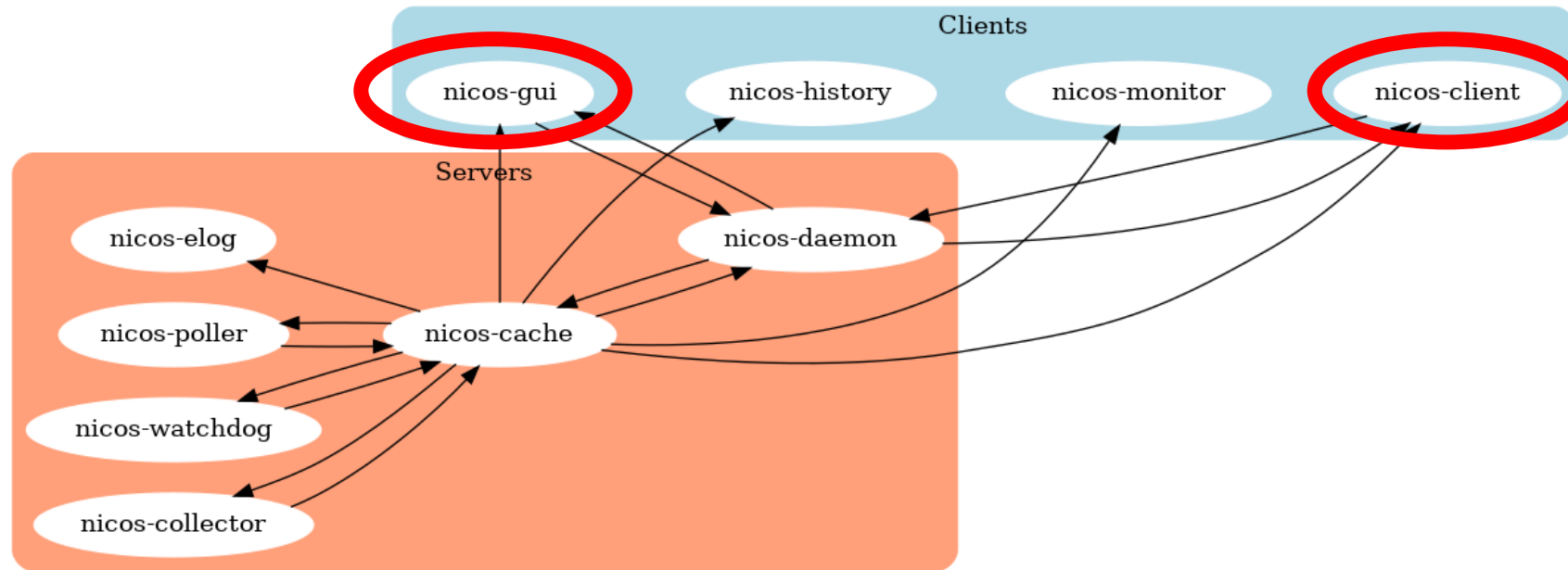
- Watchdog
 - Check for events and send out alerts (Email, SMS, Slack, ...)
 - Configurable conditions

Components



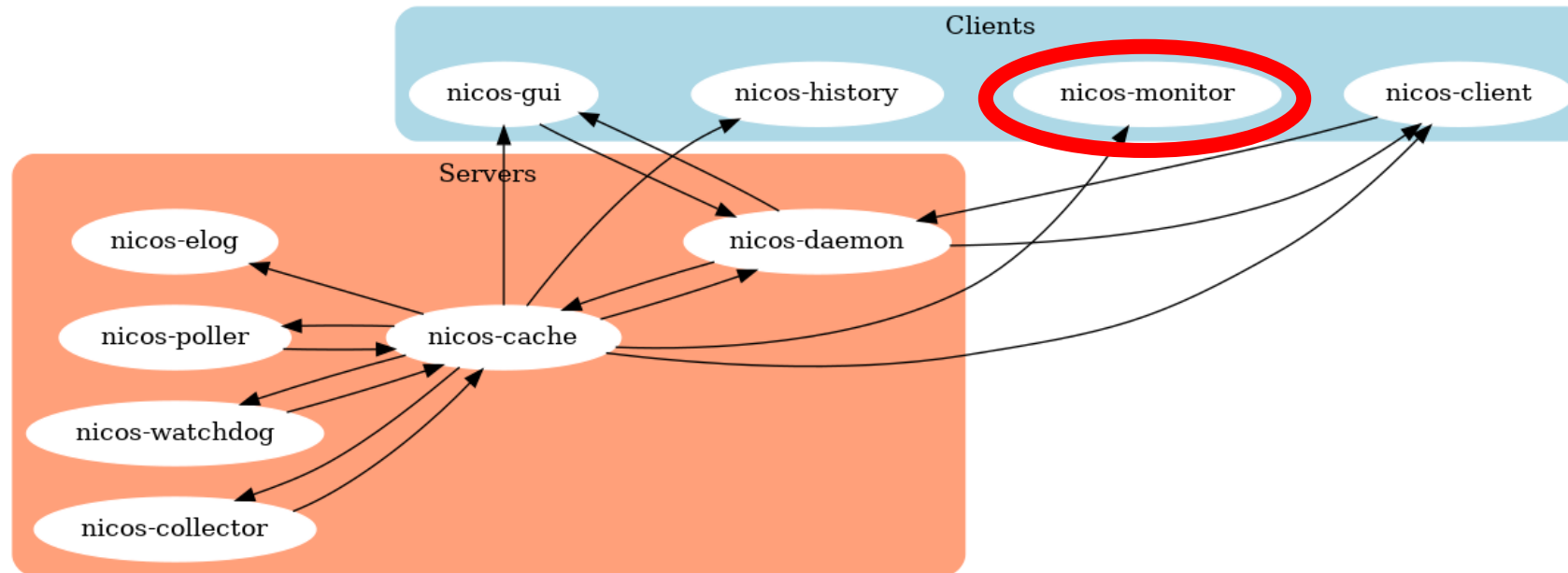
- Collector: transfer data to/from other cache(s)

Components



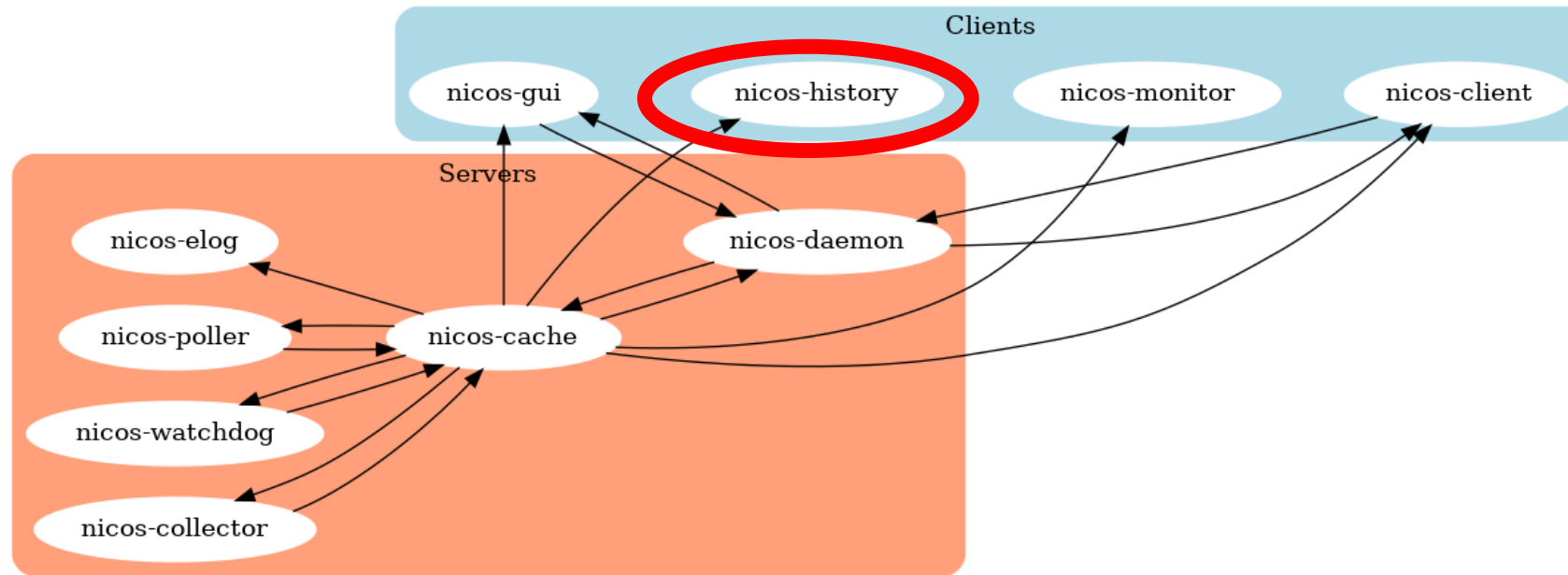
- GUI/client: user interaction

Components



- Monitor: Display status/value/parameter of devices
- Qt and HTML version

Components



- History: Display archived data depending on time

Development

Development

- Git as version control system

Development

- Git as version control system
- Test suite to catch regressions before integrate changes into source

Development

- Git as version control system
- Test suite to catch regressions before integrate changes into source
- CI with checking against test suite, static code checker, and setup check

Development

- Git as version control system
- Test suite to catch regressions before integrate changes into source
- CI with checking against test suite, static code checker, and setup check
- Code review system (Gerrit) to review, discuss, and improve changes

Development

- Git as version control system
- Test suite to catch regressions before integrate changes into source
- CI with checking against test suite, static code checker, and setup check
- Code review system (Gerrit) to review, discuss, and improve changes
- Code is freely available (GNU public license)

Development

- Git as version control system
- Test suite to catch regressions before integrate changes into source
- CI with checking against test suite, static code checker, and setup check
- Code review system (Gerrit) to review, discuss, and improve changes
- Code is freely available (GNU public license)
- Project page: <https://www.nicos-controls.org> with links to
 - Source repository
 - Issue tracker
 - Code review

Development

- Git as version control system
- Test suite to catch regressions before integrate changes into source
- CI with checking against test suite, static code checker, and setup check
- Code review system (Gerrit) to review, discuss, and improve changes
- Code is freely available (GNU public license)
- Project page: <https://www.nicos-controls.org> with links to
 - Source repository
 - Issue tracker
 - Code review
- Readonly copy on GitHub: <https://github.com/mlz-ictrl/nicos>

Some reasons to use NICOS at BNC

Some reasons to use NICOS at BNC

- NICOS is widely distributed and used at European Neutron Facilities
 - MLZ 25 instruments + 5 in preparation + several lab installations
 - PSI 16 running instruments
 - ESS ~10 ready to run installations (waiting for hardware and neutrons)

Some reasons to use NICOS at BNC

- NICOS is widely distributed and used at European Neutron Facilities
 - MLZ 25 instruments + 5 in preparation + several lab installations
 - PSI 16 running instruments
 - ESS ~10 ready to run installations (waiting for hardware and neutrons)
- NICOS is mature (more the 15 years in operation)

Some reasons to use NICOS at BNC

- NICOS is widely distributed and used at European Neutron Facilities
 - MLZ 25 instruments + 5 in preparation + several lab installations
 - PSI 16 running instruments
 - ESS ~10 ready to run installations (waiting for hardware and neutrons)
- NICOS is mature (more the 15 years in operation)
- User friendly
 - Science driven development
 - Developed from scientists for scientist
 - High user acceptance and satisfaction

Some reasons to use NICOS at BNC

- NICOS is widely distributed and used at European Neutron Facilities
 - MLZ 25 instruments + 5 in preparation + several lab installations
 - PSI 16 running instruments
 - ESS ~10 ready to run installations (waiting for hardware and neutrons)
- NICOS is mature (more the 15 years in operation)
- User friendly
 - Science driven development
 - Developed from scientists for scientist
 - High user acceptance and satisfaction
- Wide range of supported HW access layers
 - Tango, EPICS, SECoP, CARESS, ...

Some reasons to use NICOS at BNC

- NICOS is widely distributed and used at European Neutron Facilities
 - MLZ 25 instruments + 5 in preparation + several lab installations
 - PSI 16 running instruments
 - ESS ~10 ready to run installations (waiting for hardware and neutrons)
- NICOS is mature (more the 15 years in operation)
- User friendly
 - Science driven development
 - Developed from scientists for scientist
 - High user acceptance and satisfaction
- Wide range of supported HW access layers
 - Tango, EPICS, SECoP, CARESS, ...
- Open Source



NICOS in action