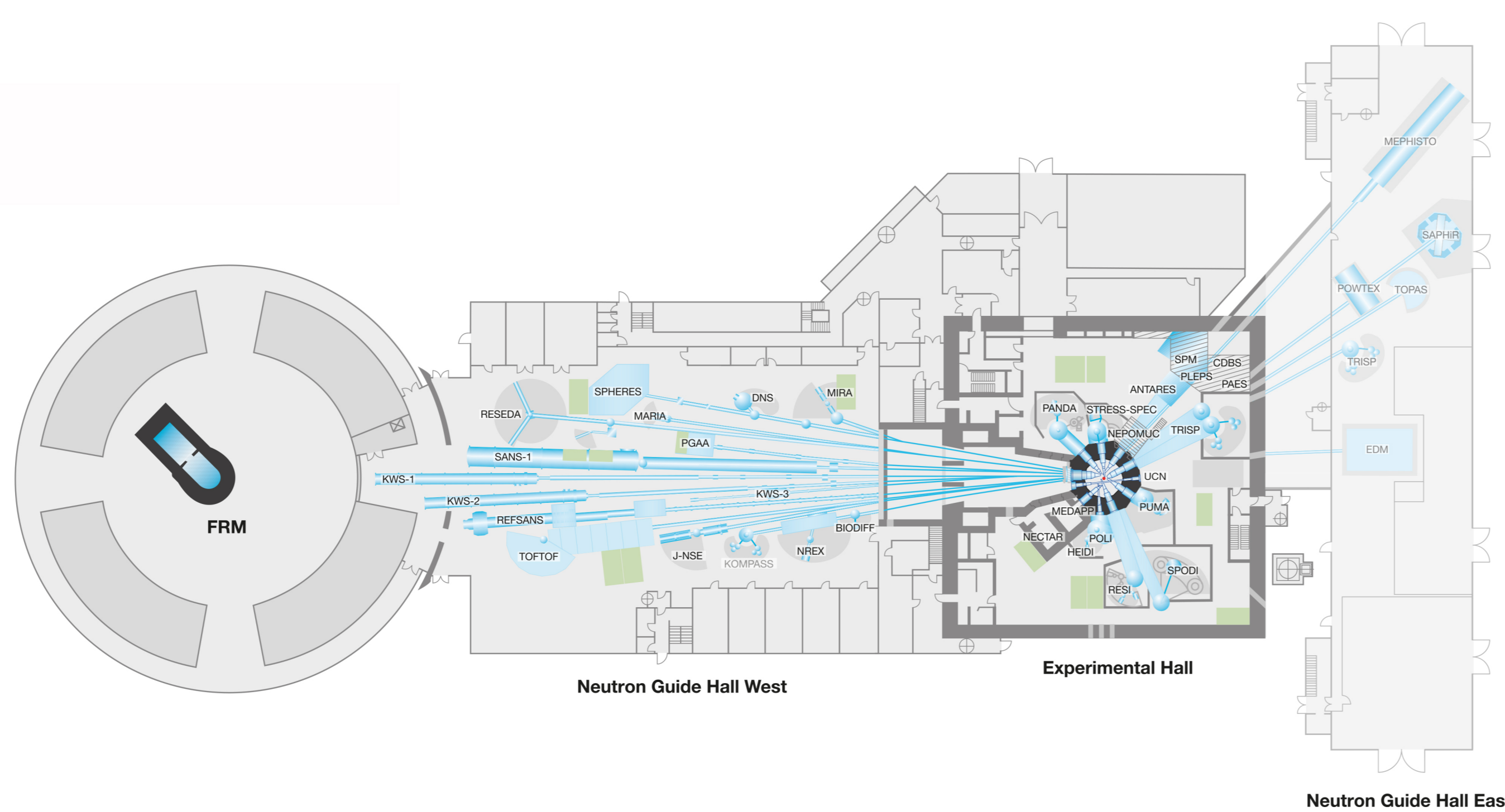


# Neutron Scattering Instrument Control System Modernization - Front-End Hardware and Software Adaption Problems

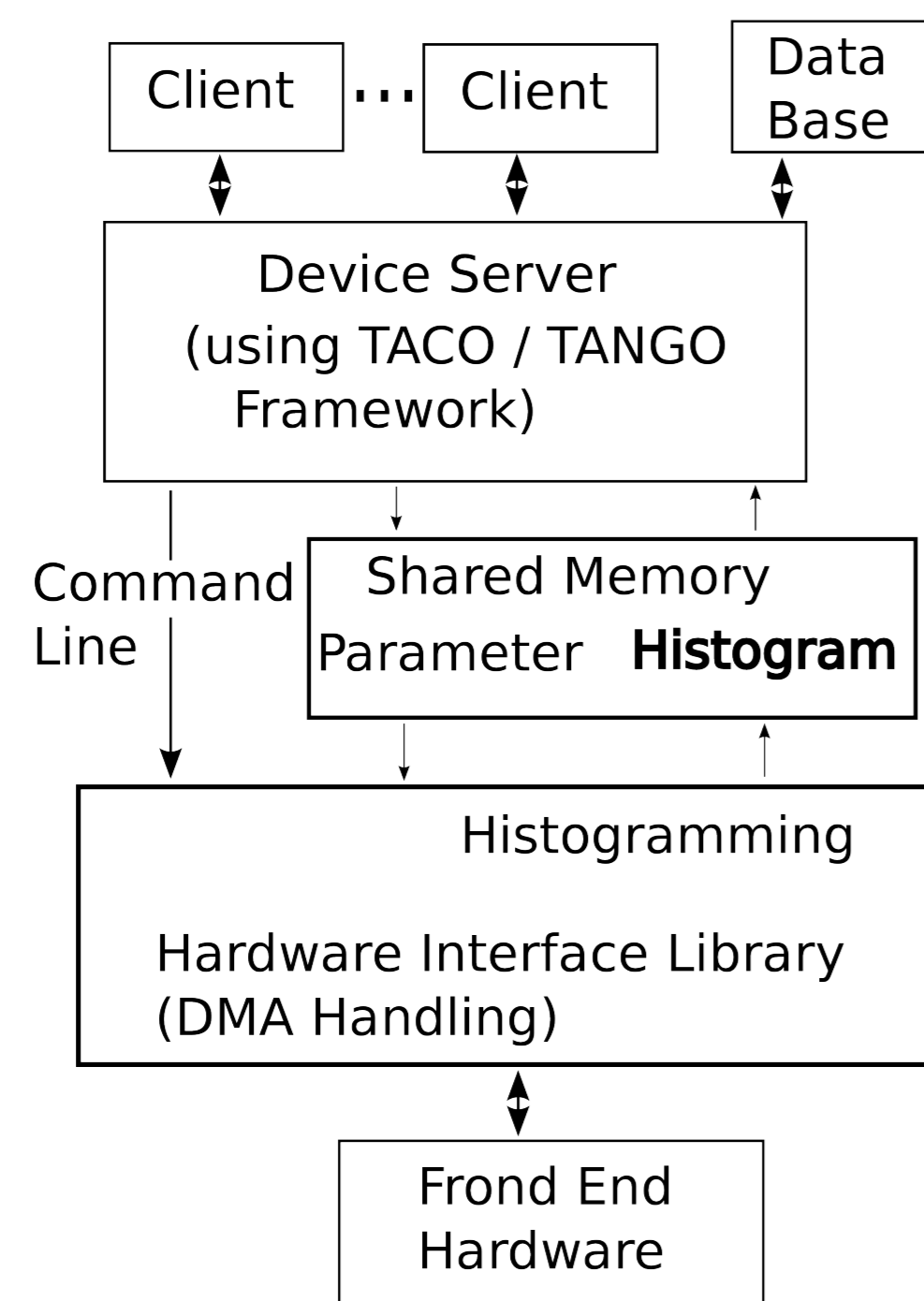
M.Drochner, L.Fleischhauer-Fuss, H.Kleines, M.Wagener, S.v.Waasen  
FZ Juelich / ZEA-2

When the FRM-2 neutron source went into operation (2002) and many instruments were moved from the closed-down Jülich reactor to the new facility, it was agreed on a choice of front-end hardware and the TACO middleware from ESRF. To keep up with software standards, it was decided recently to switch to TACO's successor - the TANGO control software. For a unified "user experience", new graphical user interface software "NICOS-2" is being developed by the software group at FRM2.

While general semantics of TACO and TANGO don't look very different at a first glance, and adaption of device servers seemed to be straightforward at first, various problems in practical operation were found. The problems were due to differences in state handling, timing behavior and error reporting. These problems, and the changes that had to be made to ensure reliable operation again, will be described.

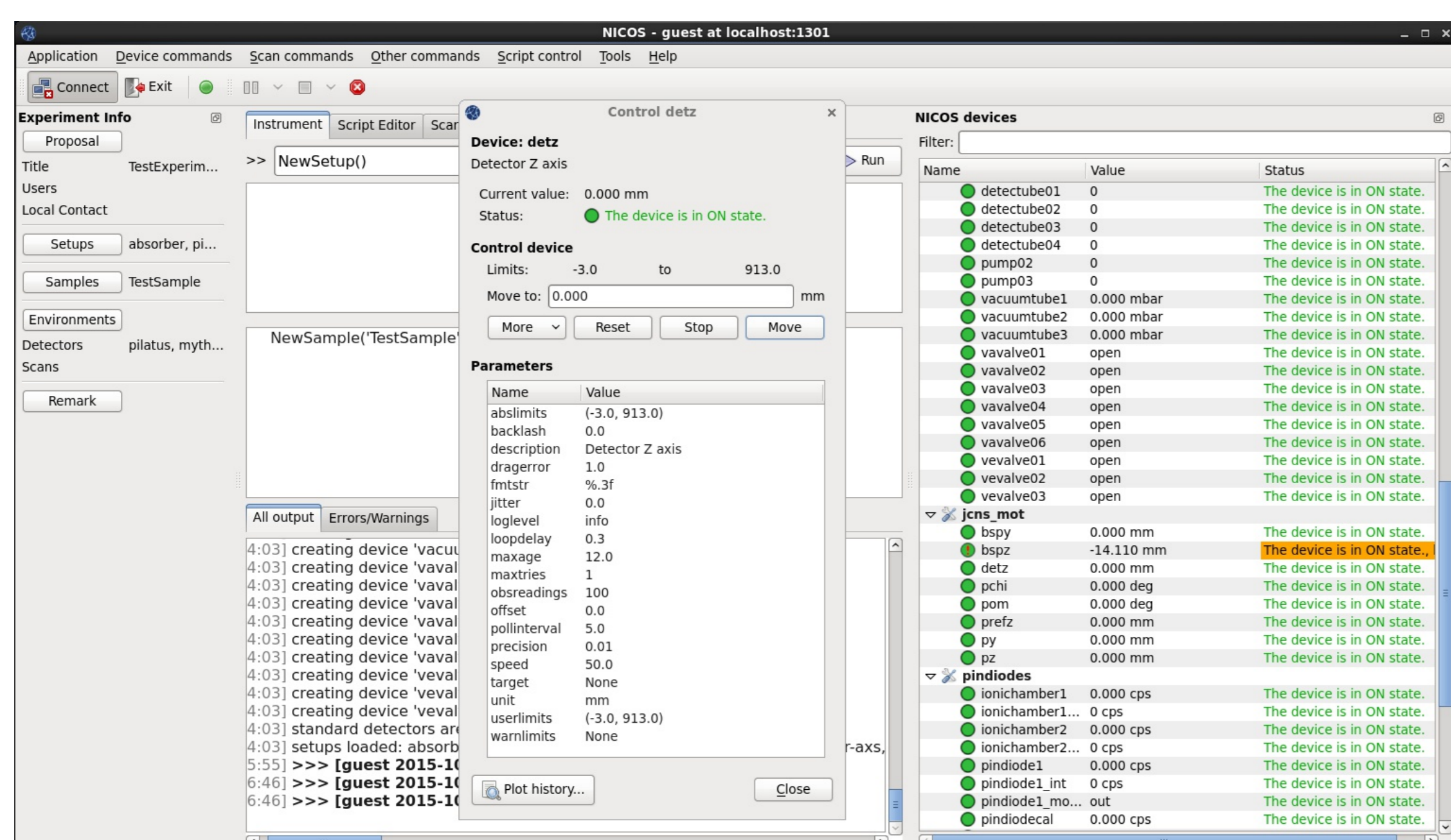


Instruments at FRM-II



## Device server for Detector

Structure of our device servers which histogram data in real time. The "Histogramming" part has been a standalone process, for easy commissioning and debugging.

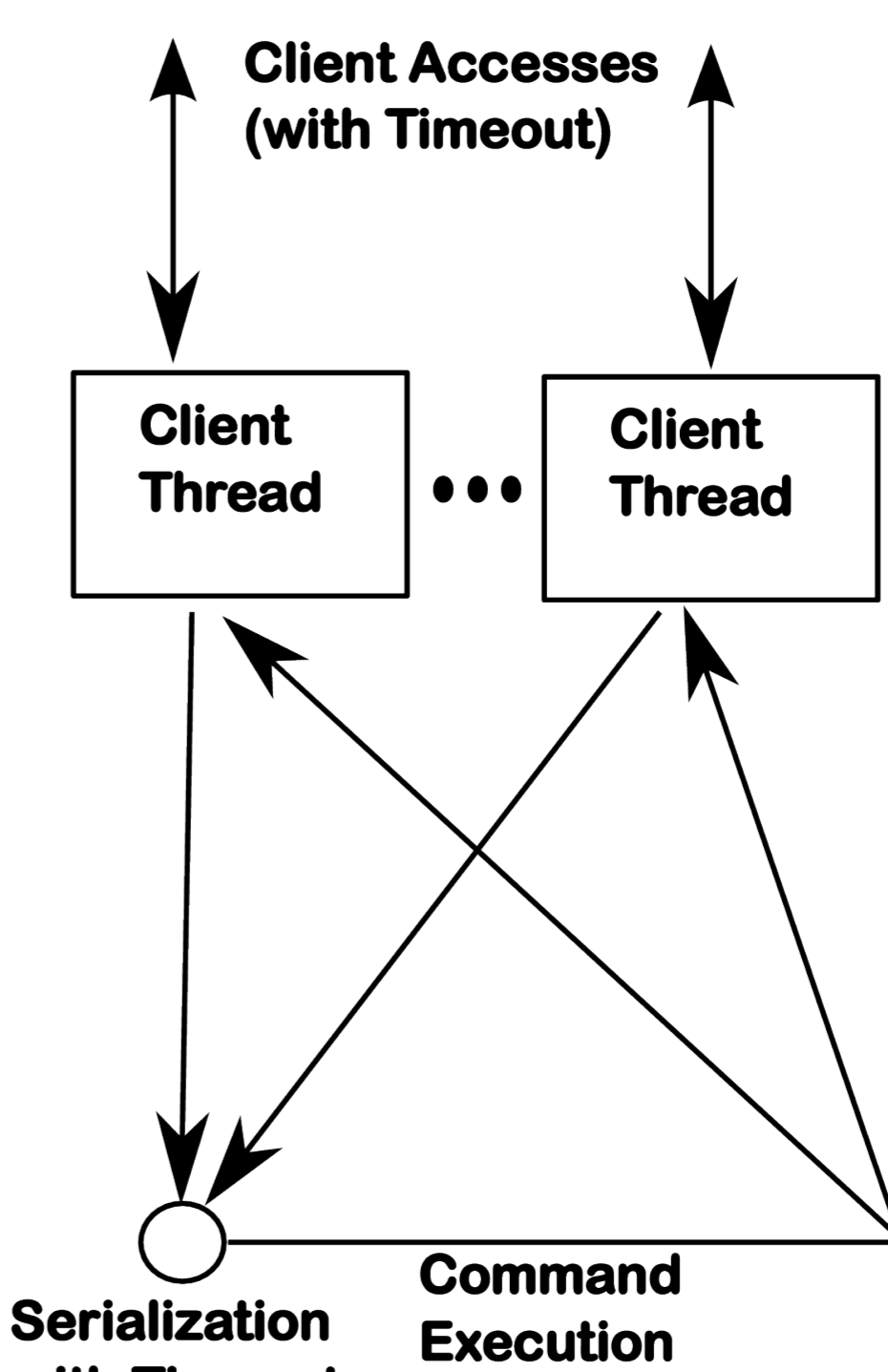


Screenshot of NICOS UI

- NICOS-2 is a Python and Python-Qt based control software. It supports TACO and TANGO for device access.
- NICOS and TANGO are used on BIODIFF, DNS, ANTARES, MIRA, POLI and others. Use on more instruments is planned.
- Front end computers and operating systems have been unified on most instruments. (CentOS)

## Problems seen:

- High load on frontend computers, permanent hard disk activity
- Random command execution errors, showing patterns depending on usage history
- Command timeouts even when TANGO timeout was increased
- "Sluggish" user interface behavior



**TANGO Timeouts**  
The serialization timeout is fixed to 3s and can not be controlled by the user.

## Analysis and Solutions:

- NICOS is a demanding UI, accesses device servers permanently from multiple threads
- TANGO does not forgive command execution times larger than the serialization timeout on accesses by multiple clients
- Front end computers which worked well with TACO might be too small for TANGO
- OS updates can cause increased resource usage (in particular memory)
- Memory shortage and OS background activity leads to paging and slowdown
- Device server commands which (in the worst case) can be slower than the serialization timeout need to be redefined to be executed asynchronously, split into "start" and "get state" parts
- Device server states can not be easily added, needs creative reinterpretation