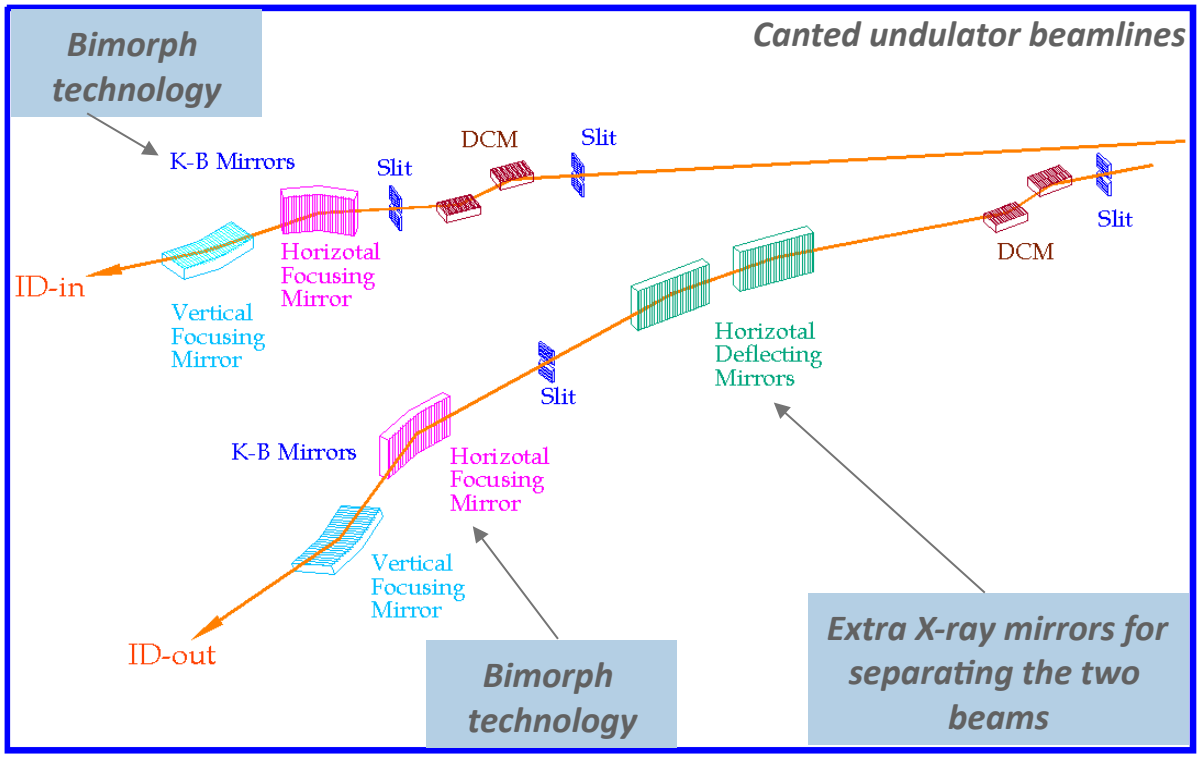
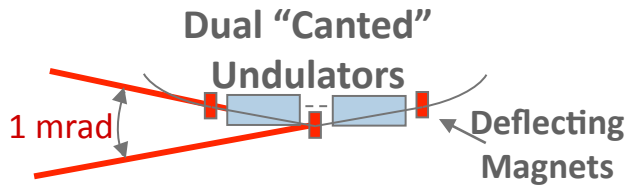
Lessons and solutions learned from marrying BluIce and EPICS

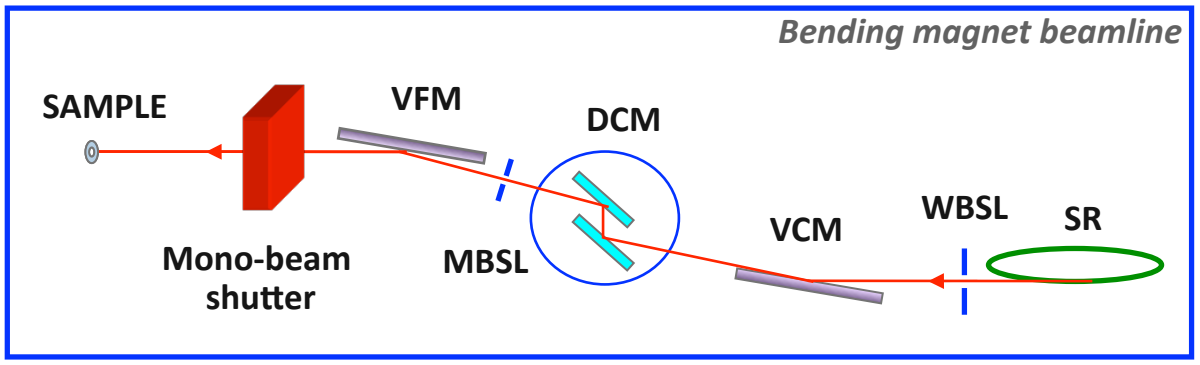
Sergey Stepanov, Oleg Makarov, Mark Hilgart, Sudhir Babu Pothineni, Derek Yoder and Robert Fischetti

General Medicine and Cancer Institutes Collaborative Access Team
(GM/CA CAT) at the Advanced Photon Source
Biosciences Division, Argonne National Laboratory

GM/CA CAT: three dedicated beamlines for MX crystallography







Hardware:

- Delta Tau PMAC motion controllers
- 4k x 4k Rayonix CCD detectors
- ALS-style sample automounters
- On-axis visualization system
- Axis and iQeye video servers
- Mini-beam collimators
- Bimorph focusing mirrors

Project started 3/2002, first mono light 6/2004, first data set 7/2004

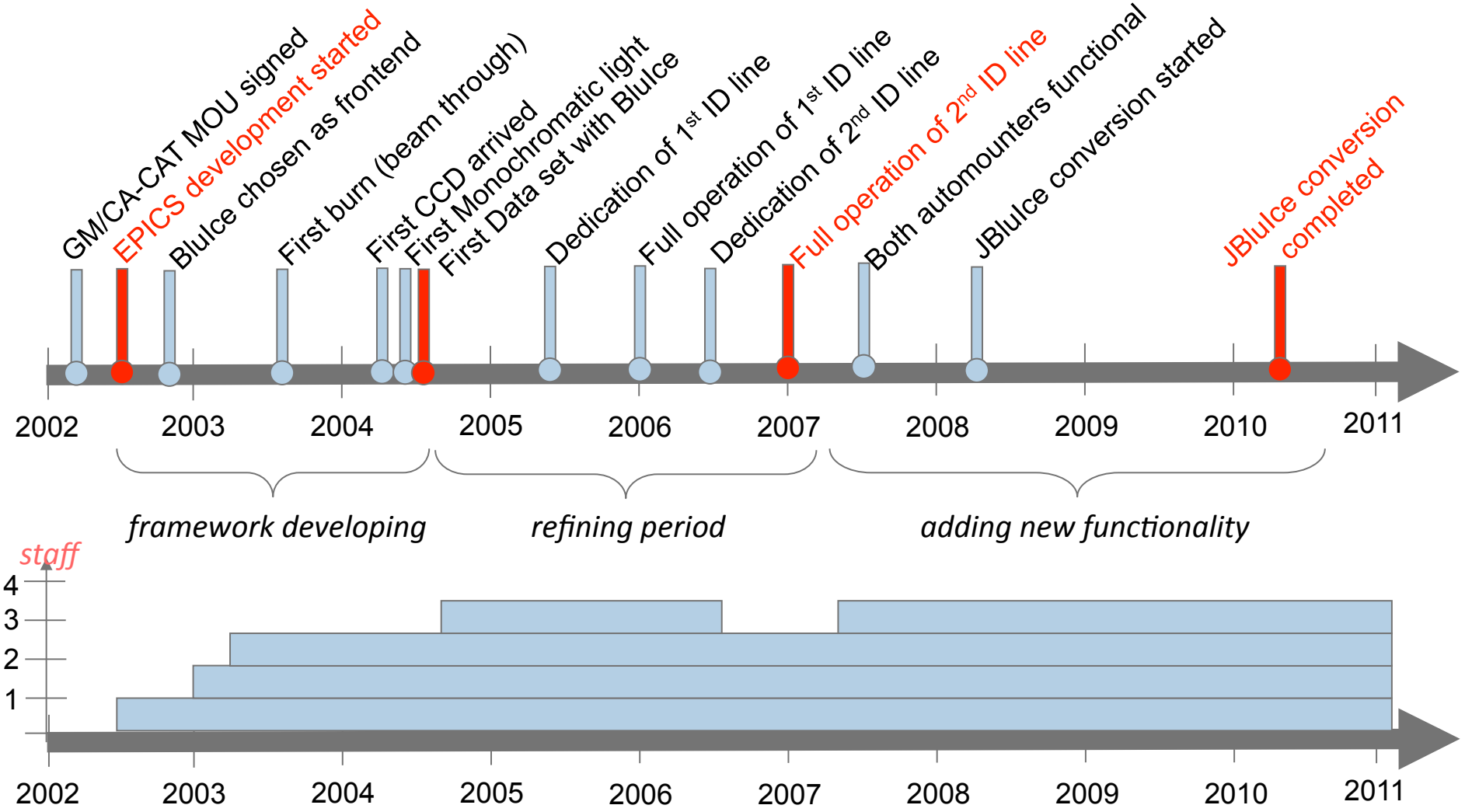
Challenges for GM/CA CAT control system

- High expectations: MX automation is the highest in SR field
- Typical experiment: 1 day; controls learning: 1 hour
- Users travel between multiple facilities: want similar interface (aka rental cars)
- Remote access is often: full automation and safeguards
- Inexperienced users: often blood test attitude – safeguards again
- Large share of sample screening: preferred to do things fast (on-the-fly scanning)
- Pipeline to data processing software
- Short terms: choice of electronics & basic controls in 1 year, GUI controls after 1 more year

Initial choices

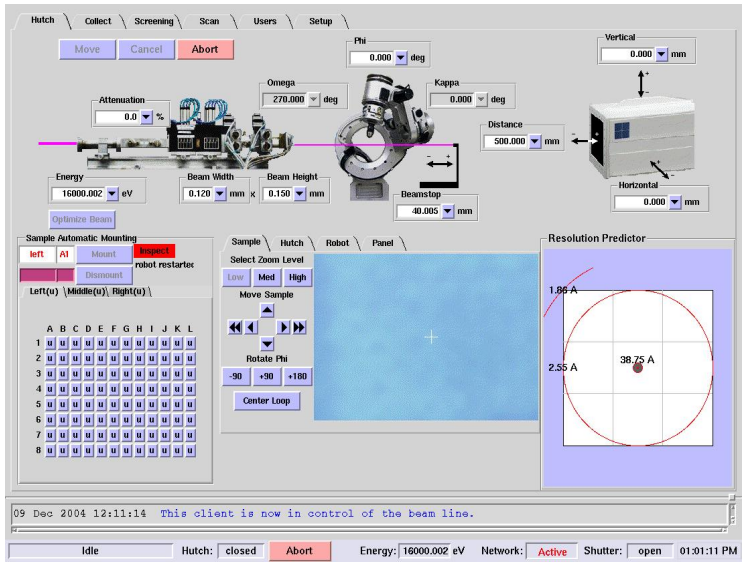
- Have both physicists and software developers in the controls team => 2+2
- Listen users what they want as an interface => BluIce
- Listen developers how to implement it best => EPICS
- Do not re-invent the wheel: make best use of what is available => BluIce & EPICS
- Stick to big projects with long lifetime and wide community => EPICS => Java/Eclipse
- Pick universal motion controller with on-the-fly scanning capability => Turbo PMAC
- Try to make software at all beamlines as close as possible => mySQL config

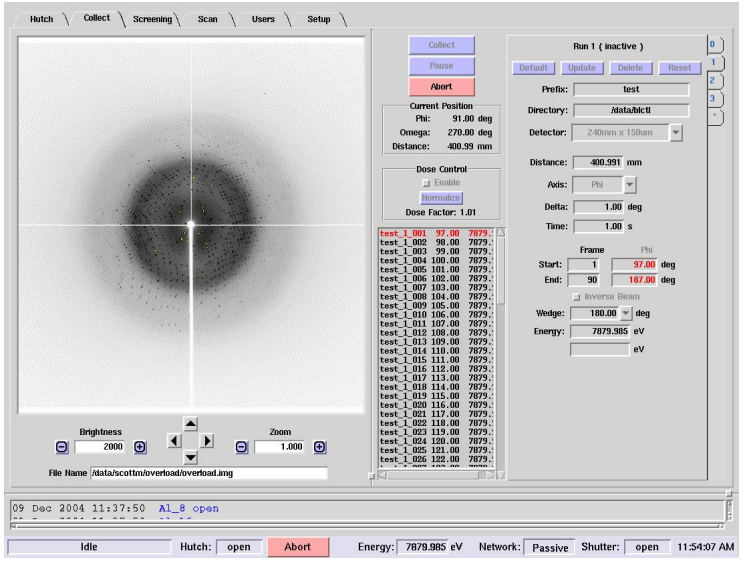
GM/CA-CAT software project timeline and staffing

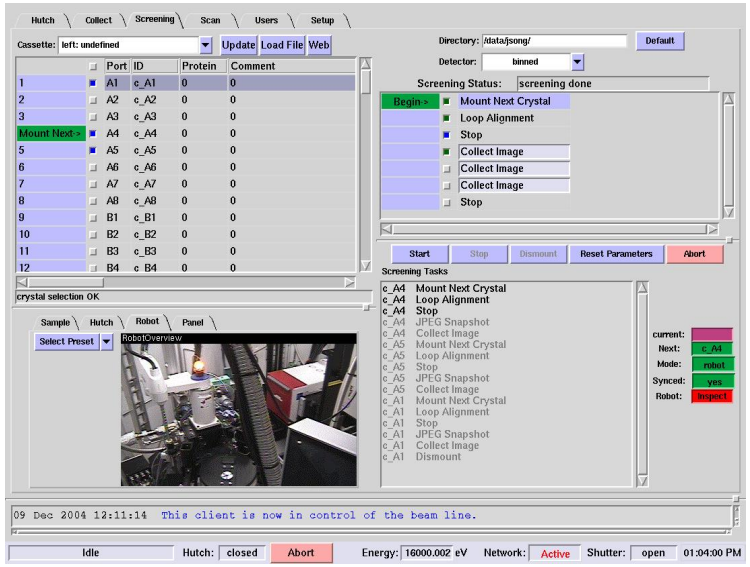


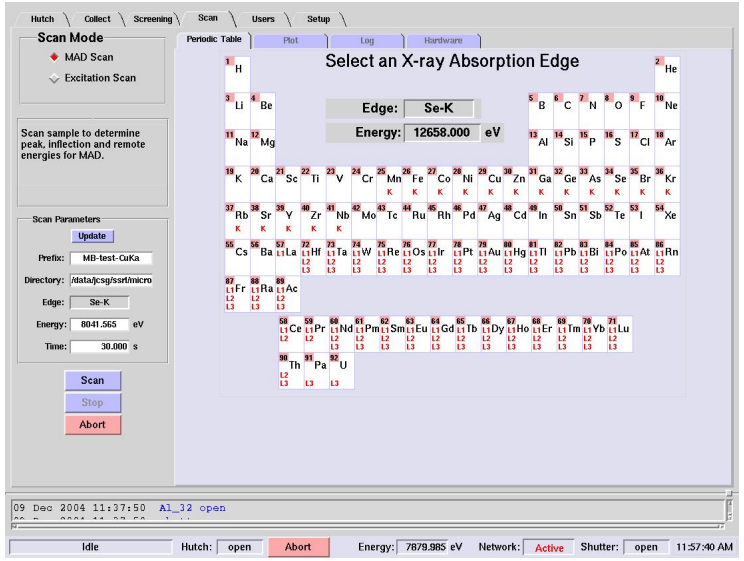
Shared developers duties: Linux/Windows system administration, SAN storage, network, user accounts (LDAP), cyber security (patching), remote access, daily support for users & staff, hardware tuning.

Quest for graphical interface: Blulce (listen what users want)









http://smb.slac.stanford.edu/

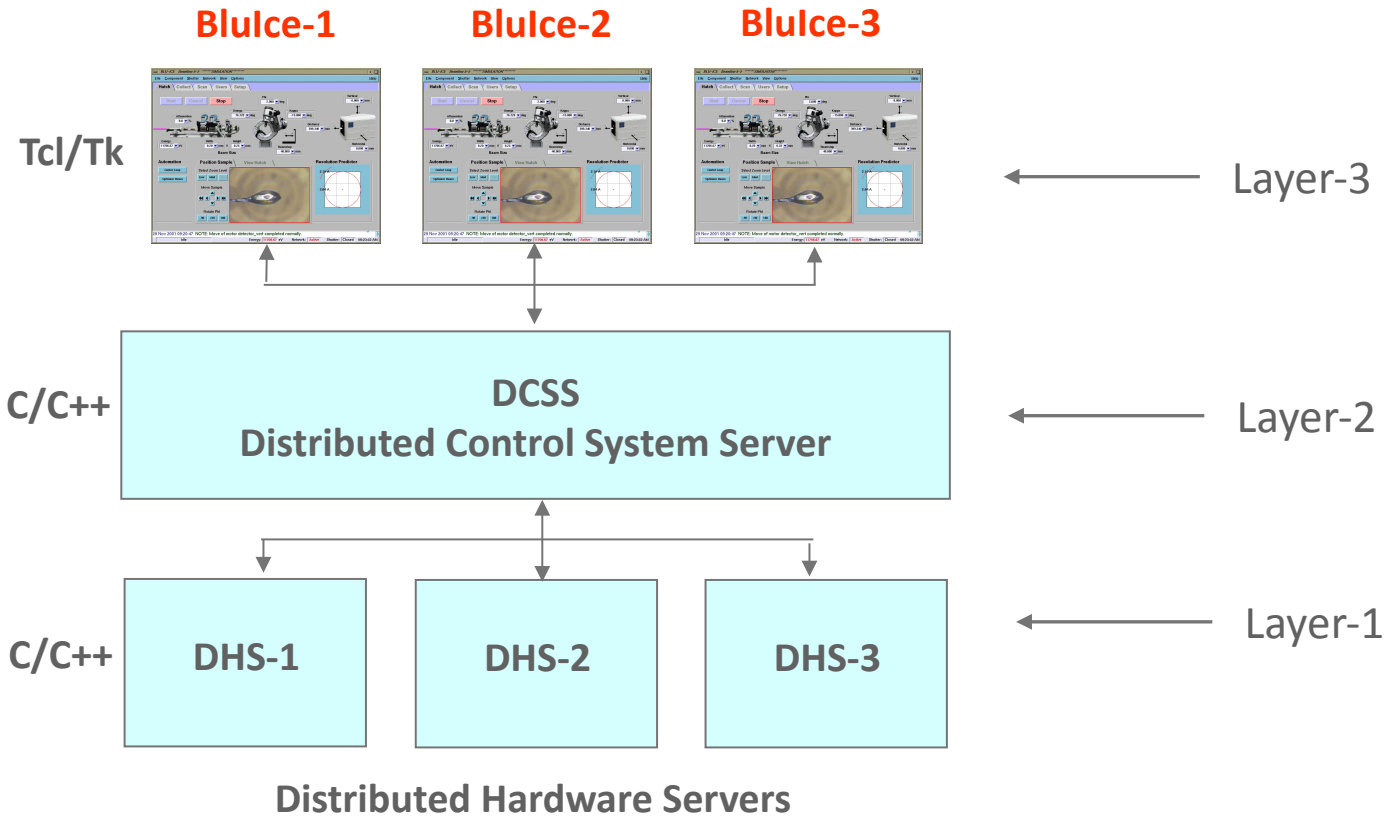
National Institute of General Medical Sciences

NATIONAL CANCER INSTITUTE

GM/CA CAT

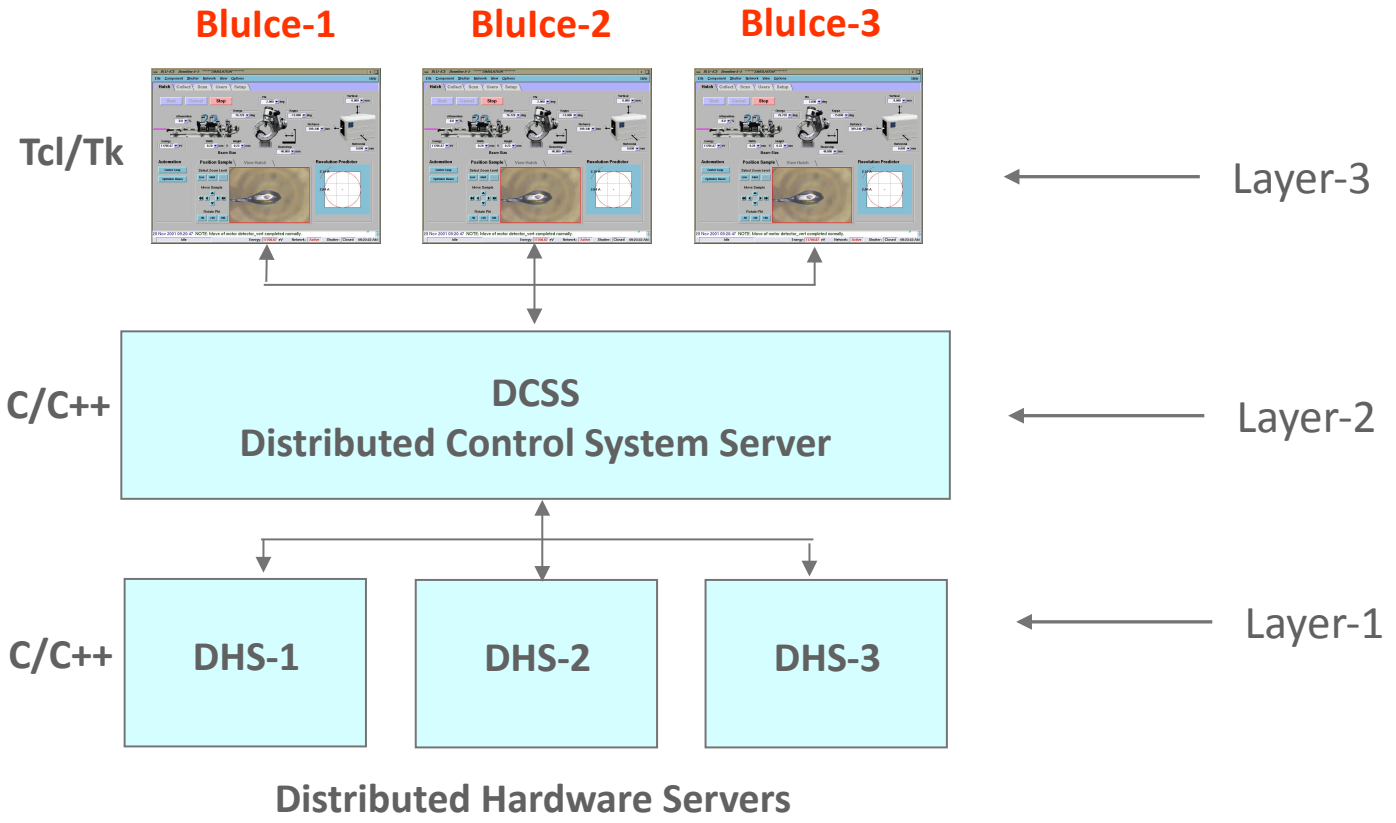
Reasons for picking BluIce

- Dedicated control system for MX crystallography, used at 7 SSRL beamlines
- Well thought tabbed interface: Hutch, Collect, Scan, Screen, and etc.
- One GUI instance is a master at a time (allowed to control the hardware)
- **Preferred by users**



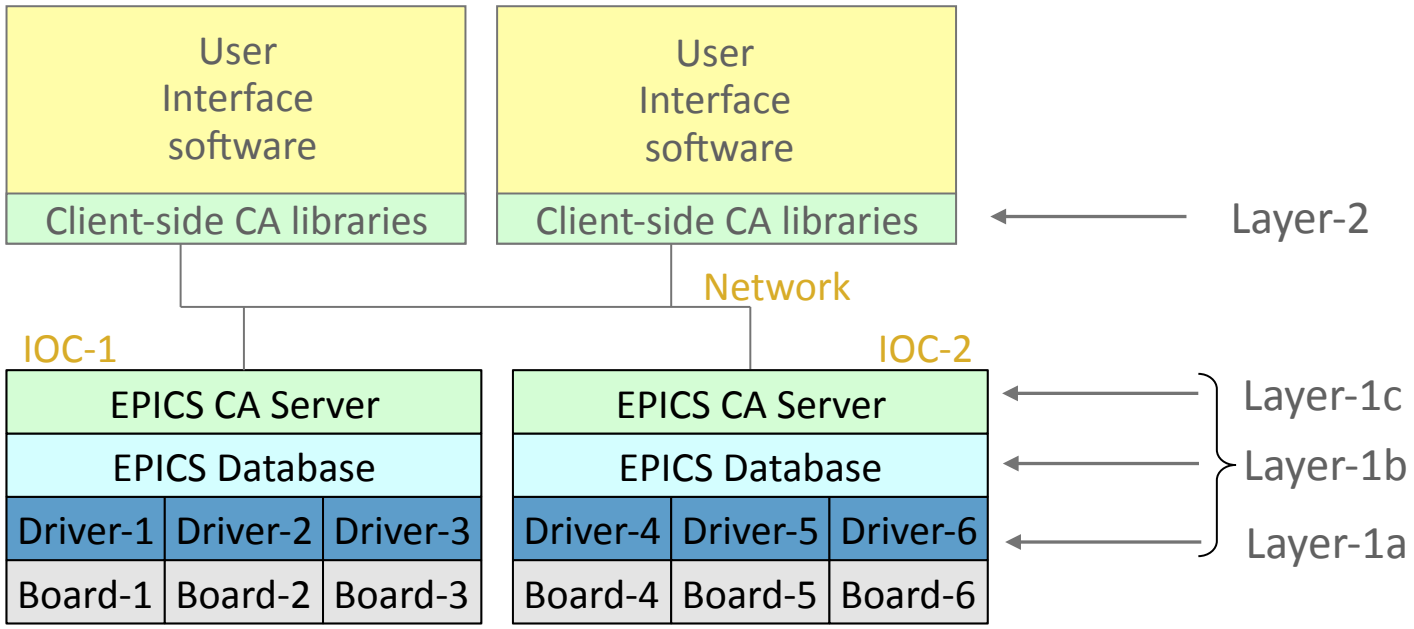
Why not using BluIce as a sole base?

- Very small selection of drivers
- DCSS makes programming easier, but masks advanced hardware access
- DCSS is a single master: not designed to monitor if others can change hardware states
- Use of Tcl/Tk for graphical interface: bad debugging tools, shortage of libraries

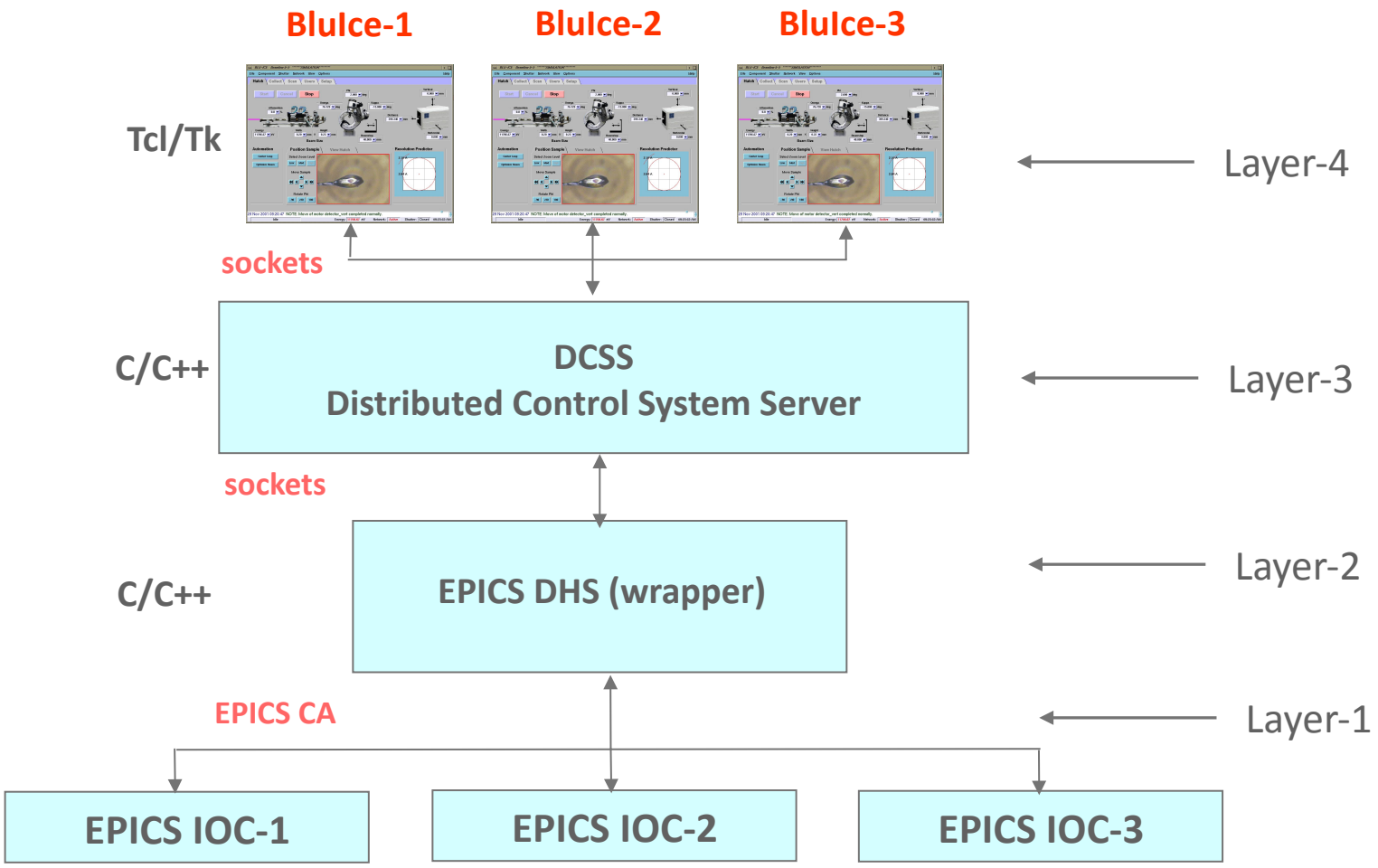


Why EPICS?

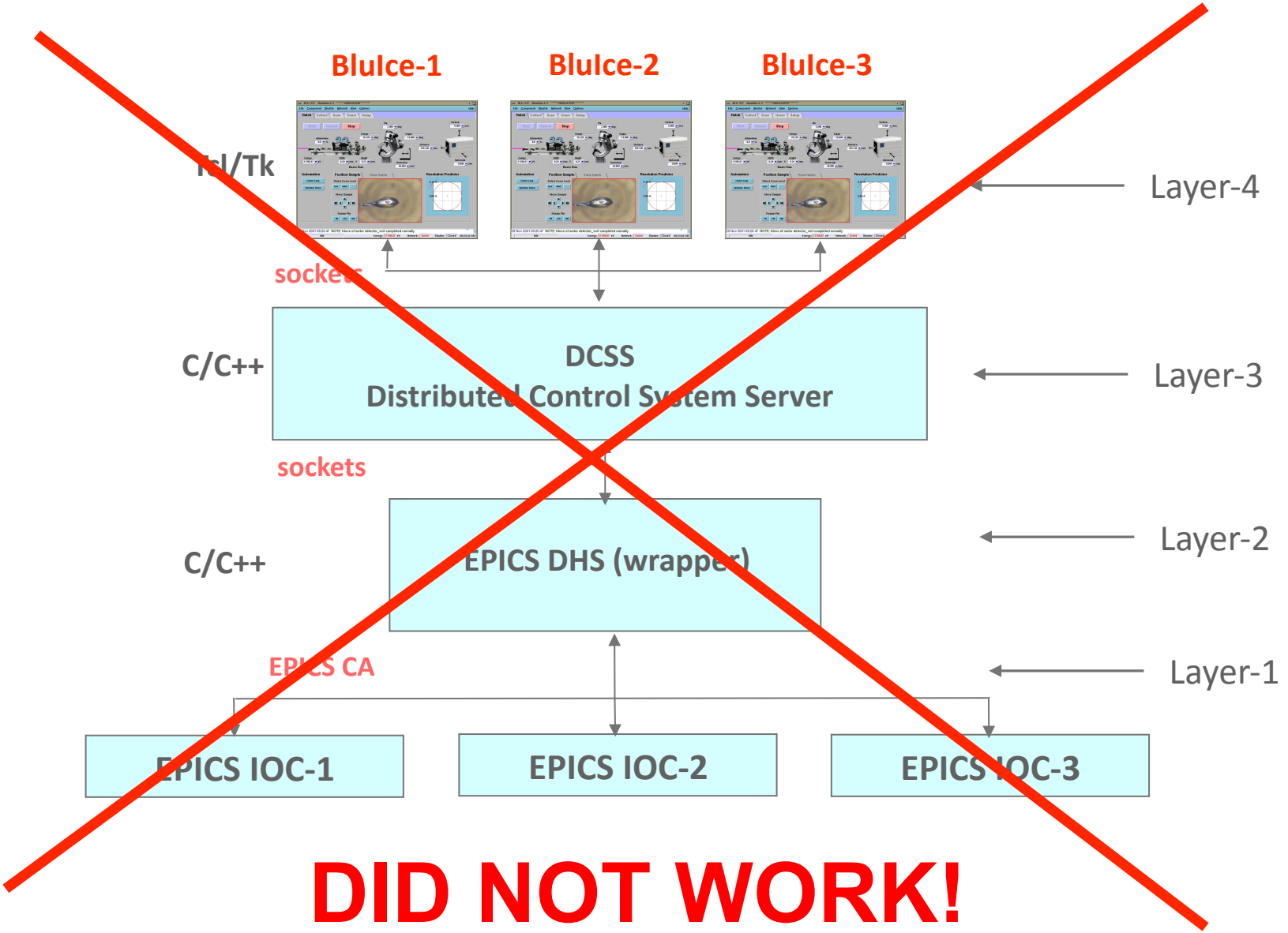
- Well established toolkit for building distributed controls
- Wide community including many synchrotrons
- Wide choice of drivers including synApps
- API for programming clients in most of existing languages
- Good choice of debugging tools (MEDM, Striptool, caMonitor)
- State Notation Language
- **Preferred by developers (local expertise)**



Choice of BluIce-EPICS interface: initial attempts

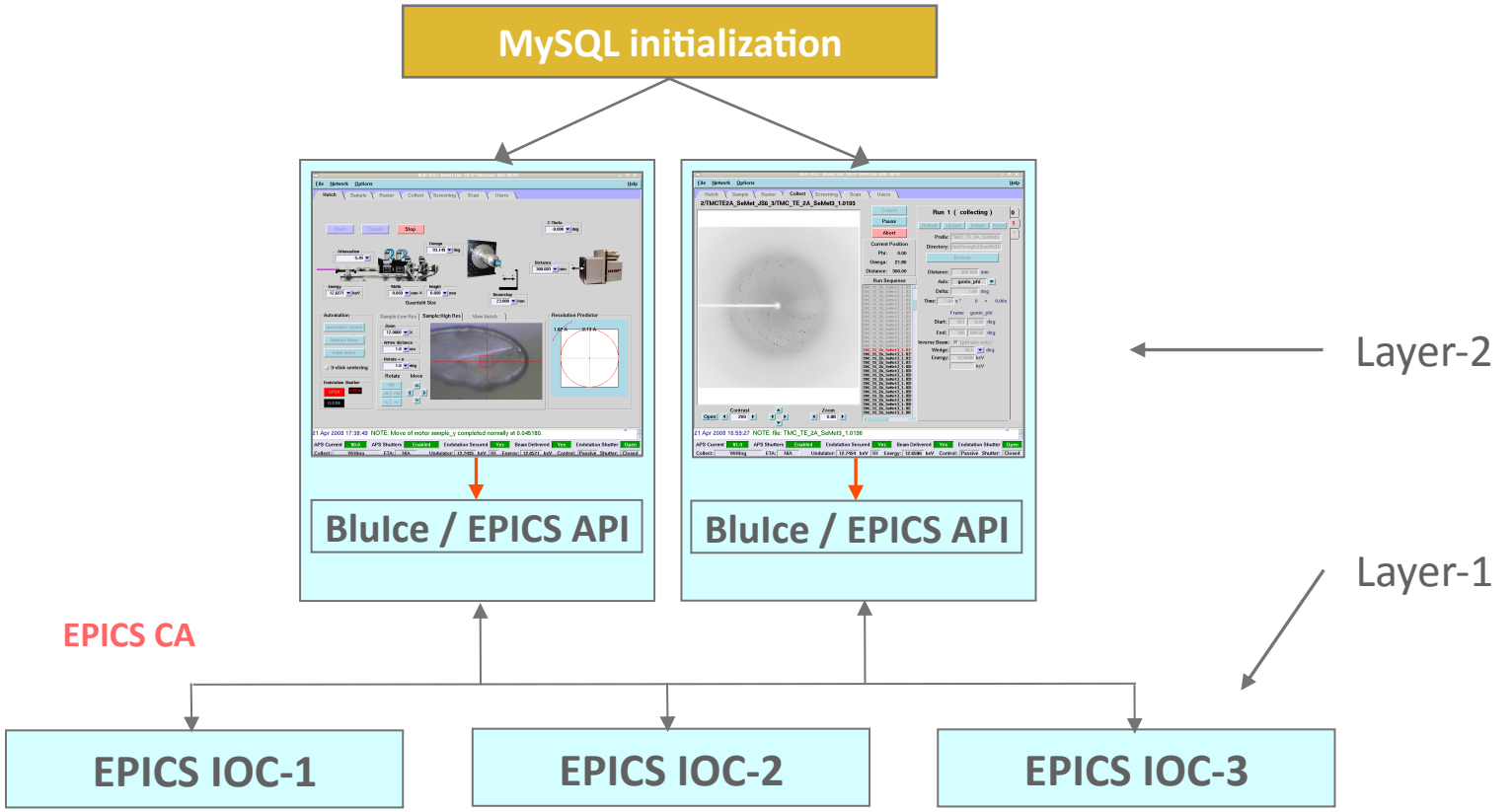


Choice of BluIce-EPICS interface: initial attempts



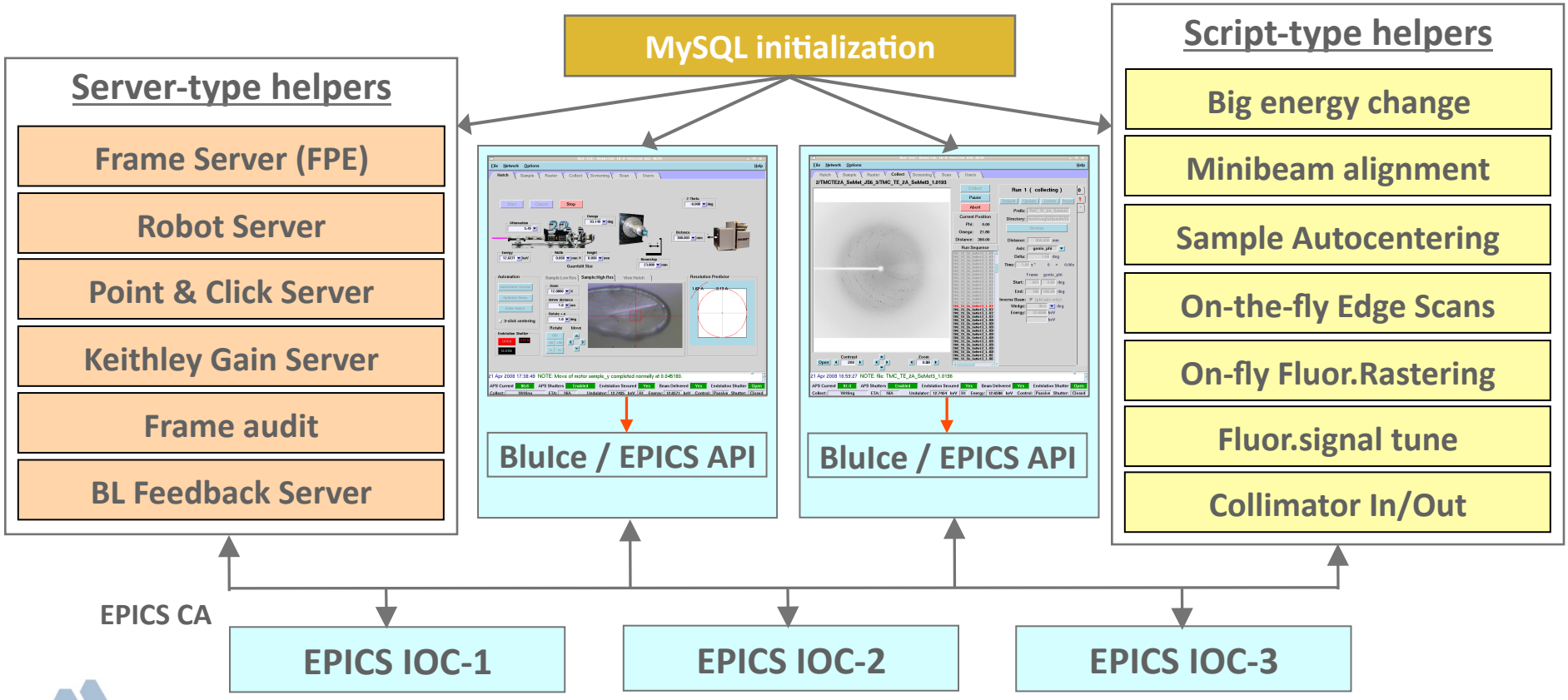
Choice of BluIce-EPICS interface: chasing speed

- Keep BluIce GUI, but convert them into standard EPICS clients like MEDM
- Communication between GUI instances via EPICS DB using callback mechanism
- Initialization via MySQL
- Less layers than in BluIce, but **heavy GUI**

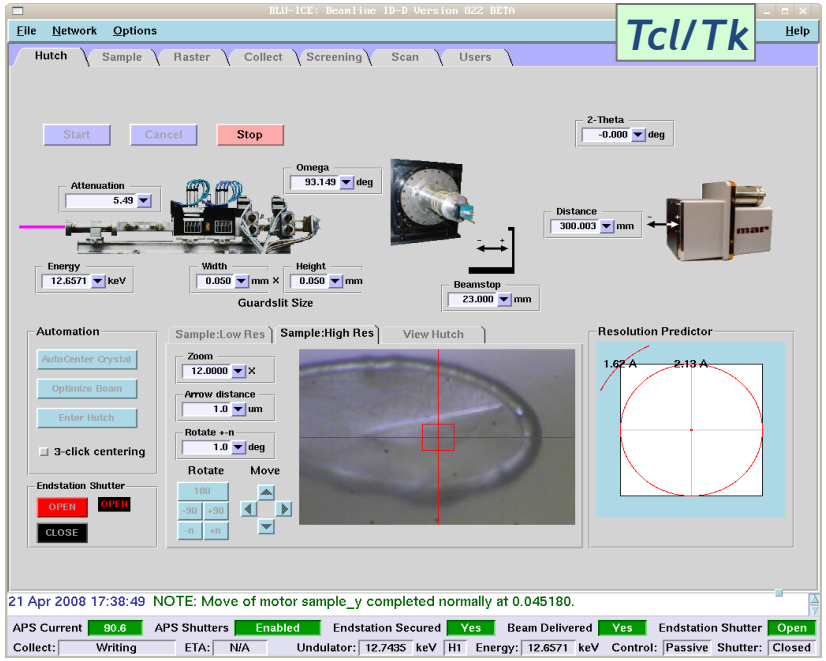


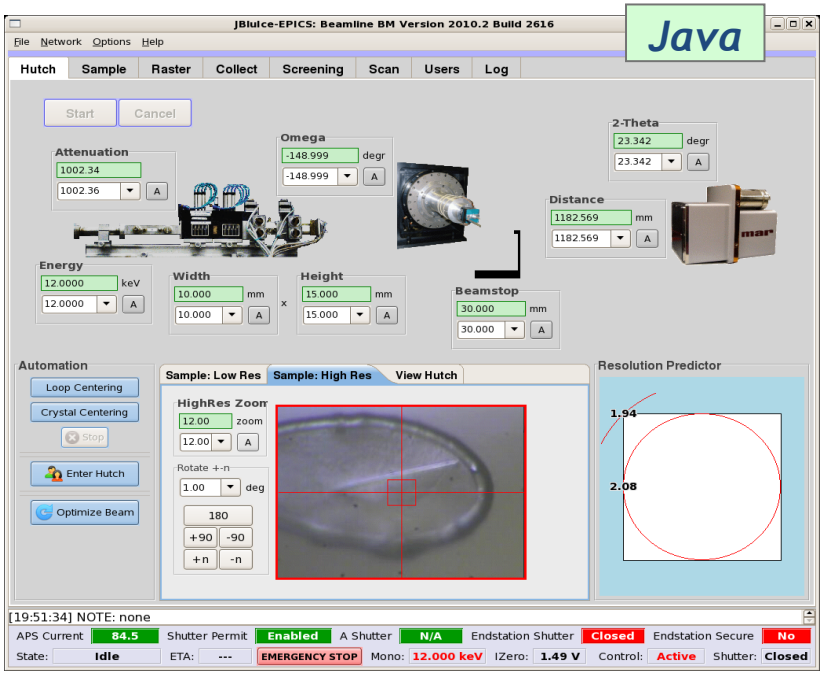
Choice of BluIce-EPICS interface: helpers

- Keep BluIce GUI, but convert them into standard EPICS clients like MEDM
- Communication between GUI instances via EPICS DB using callback mechanism
- Initialization via MySQL
- Less layers than in BluIce, but heavy GUI
- Multiple "helpers" for complex tasks; communication with helpers also via EPICS DB.
- Helpers are fully stand-alone and can be written in any languages – deploy staff effort



Transition to JBluIce in 2010 (see talk by Mark Hilgart)





Motivations for the transition: stick to a bigger project

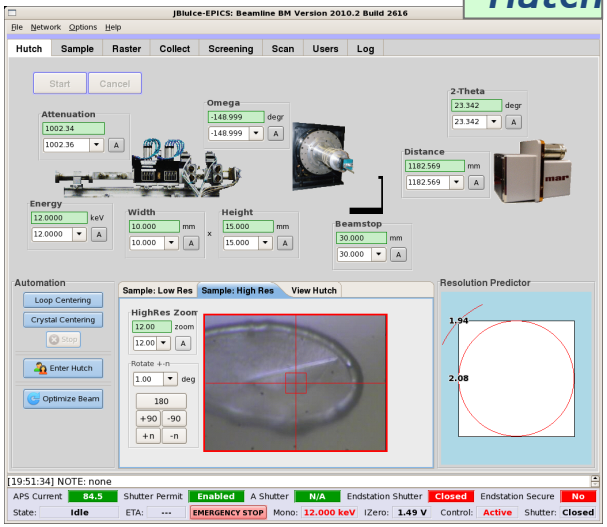
- Use extensive Java libraries (250:1 compared to Tcl)
- Simplify development and debugging through use of Eclipse RCP, the industry-leading IDE
- Secure future development by moving to a widely used and widely taught language

Gradual transition process:

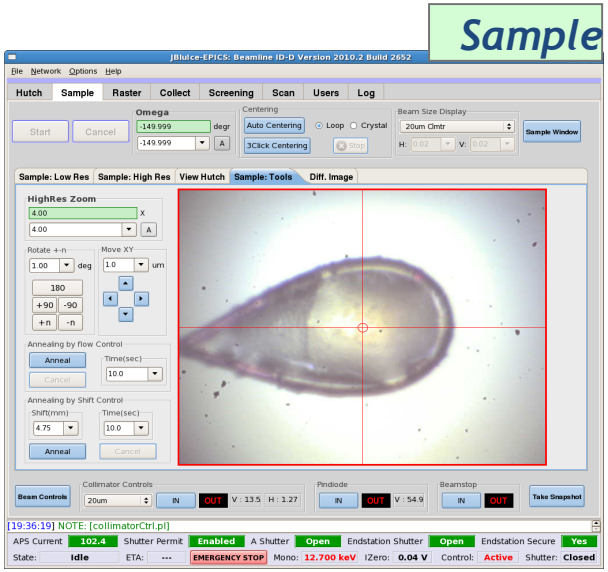
- Transition was done in 2.5 years without any interruption to users
- Possible due to all pieces communicating via EPICS (same as helpers)

JBluIce-EPICS as of today: core tabs

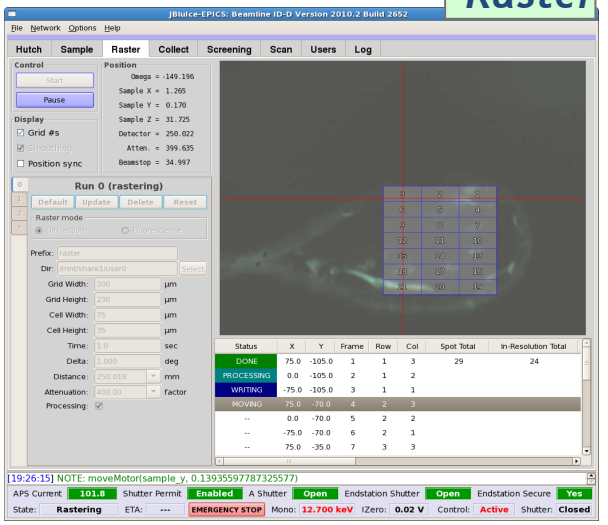
Hutch



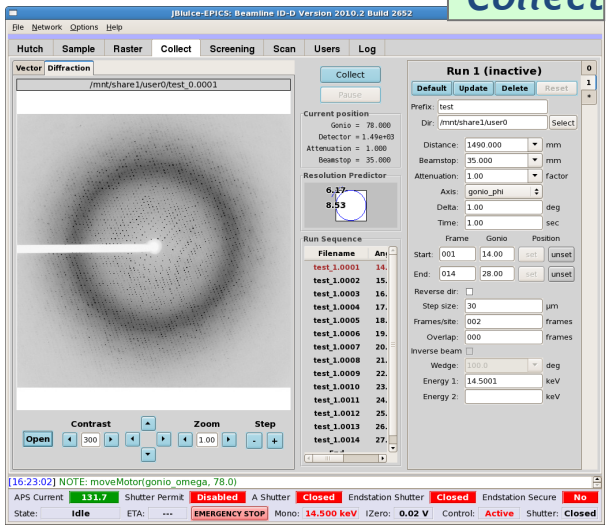
Sample



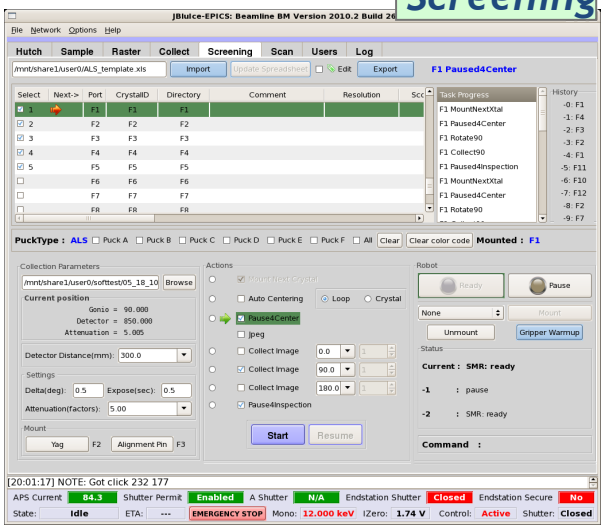
Raster



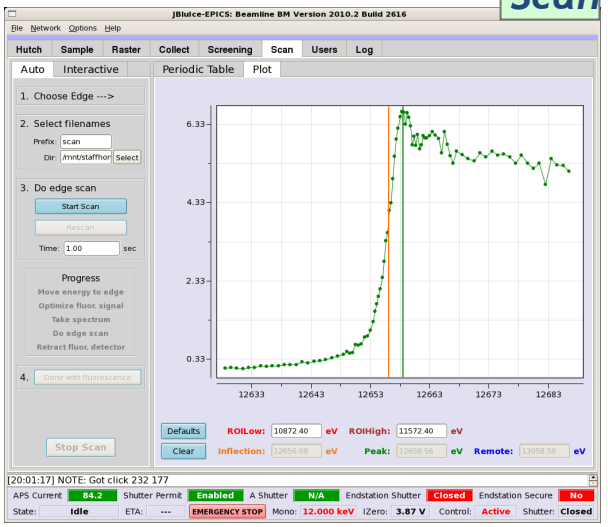
Collect



Screening



Scan



National Institute of General Medical Sciences

NATIONAL CANCER INSTITUTE

GM/CA CAT

Conclusions and plans

Worked best:

- ✓ Sticking to user-preferred interface (BluIce) and large, well-supported backend (EPICS)
- ✓ Choosing universal motion controller (PMAC)
- ✓ Making all components interact via EPICS PVs: allowed helpers & gradual Java transition
- ✓ Helpers in any language: gave flexibility and could use beamline staff hands
- ✓ JBluIce transition: easier GUI changes, easier development, more libs, secured project
- ✓ Regular listening & prioritizing users input kept driving new features (EOR forms)

Did not work:

- ❖ the seemed-most-obvious way to merge BluIce and EPICS
- ❖ listening users on how to implement features they wanted

Future plans:

- o framework is settled for now
- o more automations & safeguards, fast detectors, micro crystallography, data processing

Acknowledgements

Former members of GM/CA CAT Controls:

- *Satish Devarapalli*
- *Alex Urakhchin*

GM/CA CAT Crystallographers:

- *Michael Becker*
- *Craig Ogata*
- *Ruslan Sanishvili*
- *Nagarajan Venugopalan*
- *Ward Smith (now at NIH)*

GM/CA CAT Management:

- *Janet Smith (U. Michigan)*

GM/CA CAT has been funded in whole or in part with Federal funds from the National Cancer Institute (Y1-CO-1020) and the National Institute of General Medical Science (Y1-GM-1104).

Use of the Advanced Photon Source was supported by the U.S. Department of Energy, Basic Energy Sciences, Office of Science, under contract No. DE-AC02-06CH11357

More about GM/CA CAT at: http://www.gmca.anl.gov

Thank you !

More about GM/CA CAT at: http://www.gmca.anl.gov

APPENDIX-1: JBluIce-EPICS source code and manuals

The source code of JBluIce-EPICS is managed through Subversion.

The source code as well as the user manuals including several video tutorials are available at the GM/CA CAT web site:

http://www.gmca.anl.gov/userprogram/jbluice-epics.html

JBluIce-EPICS uses Eclipse, an open source IDE, as a development platform. The core application and the helpers depend on a number of freeware packages:

1. EPICS base, synAps, SNL, CAJ, PEZCA.
2. mySQL server.
3. Java and several extensions: MySQL Connector/J, JExcelApi, and PDFRenderer.
4. Eclipse RCP; includes Eclipse runtime, SWT, JFace, Workbench.
5. Apache commons libraries for HTTP post support: codec, httpclient, and logging packages.
6. Perl 5 or later and several popular Perl extensions downloadable from Perl CPAN archive: PDL, ImageMagick, DBD-mysql, MIME-Lite, and Perl/Tk.

JBluIce also interfaces: WebIce, XREC, DISTL, CHOOCH, ADXV; more to come.

APPENDIX-2: Remote Access

Users want:

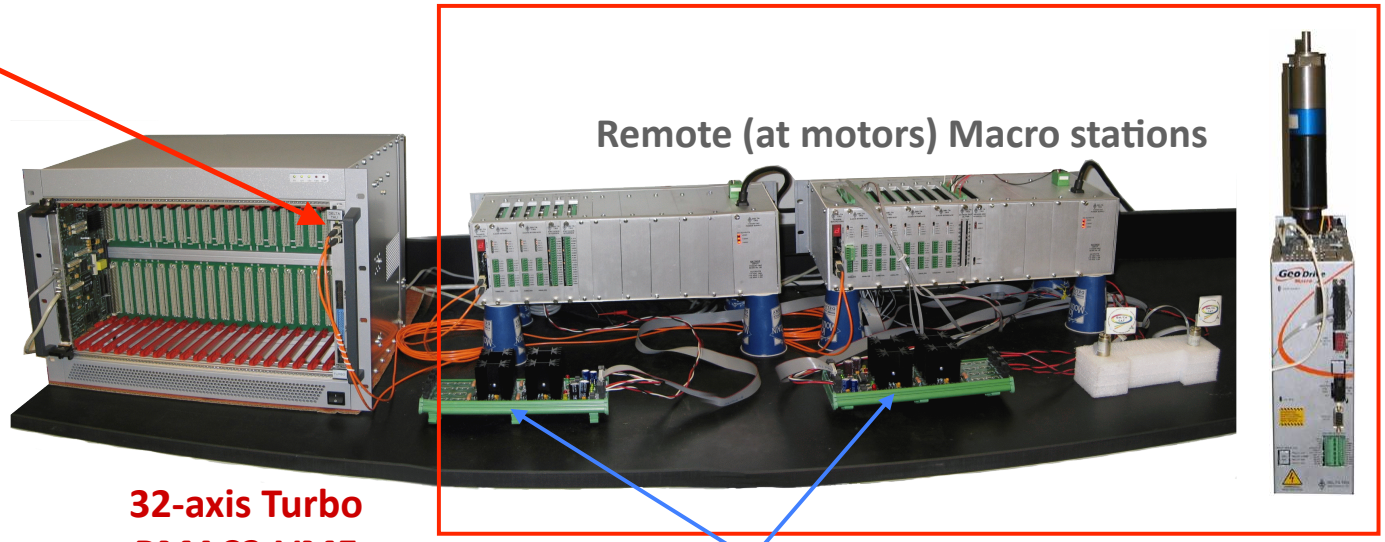
- remote data collection,
- remote data processing,
- remote data backups

We can provide:

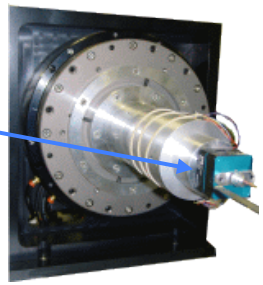
- remote data collection, but with a lot of effort & re-inventing the wheel
- remote backups (more-less easily)
- no remote data processing with some commercial software like hkl2000.
- **Solution:**
- Use NOMACHINE NX – industry standard
- Open 3 workstations per beamline: collection, processing, backups
- Extra security with IP domain in ssh config

APPENDIX-3: Choice of motion controller

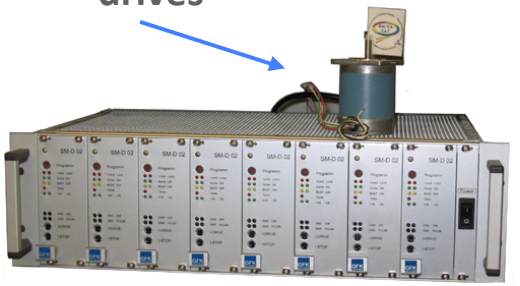
- 2-3 PMAC boards can control the whole beamline
- Works with virtually all types of motors: uniform controls
- Distributed control (fiber ring): shorter cables, less noise
- On-the-fly motion synchronization of motor assemblies
- Custom motion programs
- Analog & Digital IO



Air Bearing Gonio (brushless servo) & piezo XY head



Stepper drives



APPENDIX-4: EPICS developments

- Most parts of EPICS layer were available out of box. This included:
  - ✓ The basic infrastructure for booting controllers (MVME & Linux)
  - ✓ EPICS DB managers and Channel Access servers
  - ✓ A number of device drivers including support for Struck multiscaler, Canberra MCA, Acromag ADC & Digital I/O, Systran & Xycom DAC and etc.
- Some EPICS drivers were developed in house by extending existing drivers:
  - ✓ CCD detector interface was updated to the latest Rayonix remote interface
  - ✓ Delta Tau PMAC driver was extended to work with Turbo PMAC2
  - ✓ HYTEC-8402 DAC driver was supplied with option to do on-the-fly DAC scans
- Several State notation servers were implemented (helpers):
  - ✓ Frame Processing Engine
  - ✓ ALS-style sample automounter server
  - ✓ Keithley 428 Current Amplifier server

APPENDIX-5: BluIce developments

- Initial development: converting BluIce calls into CA calls in EPICS API
- Further developments: mostly driven by users: new tabs, new buttons
- Some in core GUI (diffraction rastering, vector collection, attenuation) and some in helpers
- Incorporating communications with new EPICS devices and helpers
- Support for iQeye cameras and new image server; interface to WebIce
- Adding safeguards all over