Argonne NATIONAL LABORATORY

Introduction to EPICS

Andrew Johnson, AES-SSG

Outline

- What is EPICS
- History
- What does it do
- The EPICS Community
- The APS EPICS Ecosystems
- Alternative Ecosystem Choices

What is EPICS?

What is EPICS?

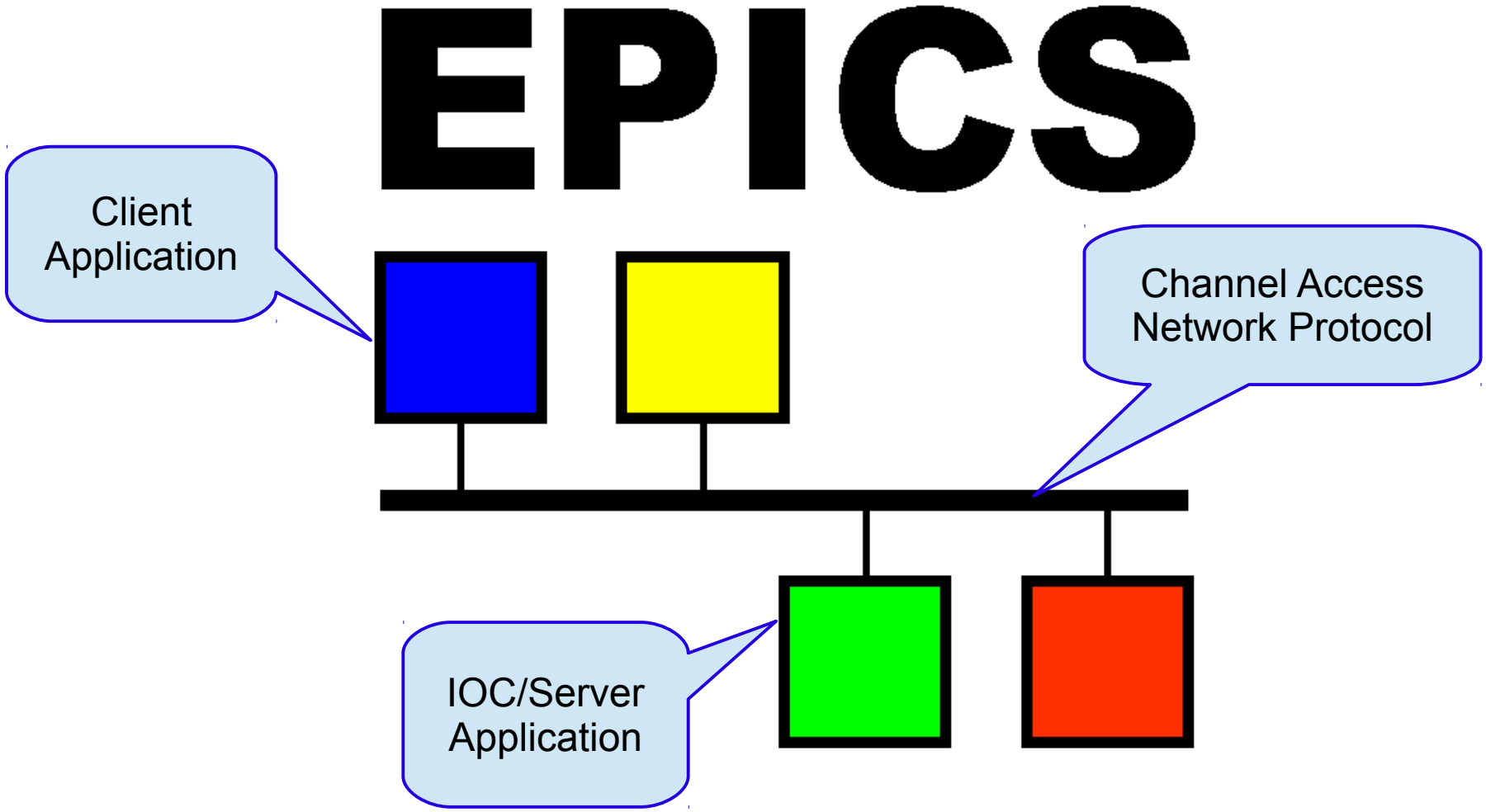
- A Collaboration
- A Control System Architecture
- A Software Toolkit

1: EPICS Collaboration

1: EPICS Collaboration

- International group of laboratories and companies that use and develop EPICS
  - Many more than represented by logos on the previous slide
  - At least 24 countries have one or more labs using EPICS
- Mailing list 'tech-talk' for users, technical questions, bug reports etc.
  - 700+ subscribers
- Organize twice-yearly meetings to discuss progress and new projects
  - Location rotates between Europe, America and Asia/Pacific
  - Open to anyone, typically 100 – 150 attendees
  - ~75 presentations over 3 days, plus 1-2 days for training and developer meetings
- Additional collaboration occurs continually between users and developers
  - Some labs share software development effort for new features and tools
  - Others assist in finding bugs, porting to new OS's, answer questions etc.

2: EPICS Control System Architecture



2: EPICS Control System Architecture

- Servers give Clients access to named Process Variables (PVs)
  - Each PV holds a piece of data associated with the machine
    - Status, read-back, set-point, parameter etc.
  - Clients only need to know a PV's name to access it

- Example PV names and values from the APS Accelerator
  - S35DCCT:currentCC 100.2 mA
  - SRlifeTimeM 328 minutes

- As well as a value, a PV has other attributes, e.g.
  - Engineering units string
  - Timestamp
  - Alarm status and severity
  - Operating range (Low .. High)
  - Control limits (Low .. High)

3: EPICS Software Toolkit

Client Programs

MEDM ALH StripTool caget
CS-Studio Channel Archiver
EDM Perl epicsQt Python C#
Tcl/Tk SDDS Matlab Kryten

Channel Access

IOC
IOC
CAS

IOC
IOC
IOC

IOC Software

PV Database State Programs
StreamDevice AsynDriver
Timing Triggers Control Loops
AreaDetector Motor Drivers
Soft Glue Modbus Autosave

Network Attached Devices

Commercial Instruments

Custom Chassis/Panels

Technical Equipment

CA Server Application

Process Variables

3: EPICS Software Toolkit

- 'Installing EPICS' does not create a control system
- The EPICS Software Toolkit consists of hundreds of pieces of software
  - Some are individual programs for performing specific tasks
    - Creating and running Graphical User-Interfaces, archiving data, monitoring alarms etc.
  - Others are code libraries for monitor or control of specific kinds of hardware
    - IOC device and driver support for COTS equipment (PLCs, oscilloscopes etc.)
  - Others allow common programming languages and tools to interface with EPICS
    - C/C++, Perl, Python, Java, Matlab/Octave, C#, LabView etc.
  - Almost all EPICS code is Open Source (freely usable & modifiable)
    - EPICS Open License is BSD-like; GNU GPL is also used
- An EPICS control system can be made by choosing and configuring existing software
  - Writing code is often only necessary to fill gaps in the available tools

History

History

- The Accelerator Automation Application Toolkit Workshop was held at LANL in 1988
  - Brought together accelerator controls people from around the world
  - LANL was building the Ground Test Accelerator [GTA], wanted advice
  - Part of Ronald Reagan's Strategic Defense Initiative (Star Wars)
- The workshop set development direction for the LANL controls group
  - GTA Control System [GTACS] subsequently developed by Bob Dalesio, Jeff Hill et al.
- APS Control System development started at ANL in 1989
  - Marty Kraimer was sent to work at LANL with the GTACS team for 6 months
- Development continued afterwards on improved portability and extensibility
  - The software was renamed EPICS and presented at ICALEPCS-'91
- LBL (Steve Lewis), SSC (Dave Gurd) and CEBAF (Chip Watson) joined in 1992
- Commercial licenses to sell EPICS 3.8 were negotiated with 3 companies
  - Enhanced versions were sold for process control by Tate & Kinetic Systems
- Licenses for collaborators were free but required a legal agreement with LANL

History

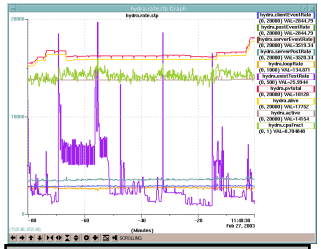
- The first EPICS website came online in August 1993
  - APS was a very early adopter of HTML & the WWW
- Code development continued, adding features, robustness and portability
  - Binary database files were replaced by ASCII in R3.11
  - New hardware support continued to be added
  - EPICS community continued to grow as new sites joined
- Until EPICS R3.14 in 2002 the IOC (server) could only run on VxWorks
  - Marty Kraimer, Jeff Hill & Janet Anderson made it portable
  - Support for RTEMS was added by Eric Norum at SAL (now CLS)
- In February 2004 EPICS was released under the EPICS Open License
  - Open Source license, approved by DOE
  - Core software is freely redistributable, may be used by anyone
- We have no recent records of who uses EPICS, or what for
  - 170 licensees under the previous license agreement (2004)
  - 2000+ downloads of EPICS Base 3.14.12.x from APS from 2013-09 to 2014-08

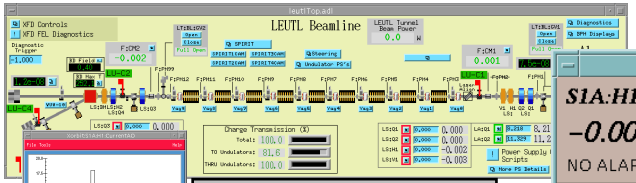
What does it do?

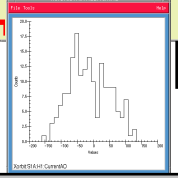
What can EPICS do?

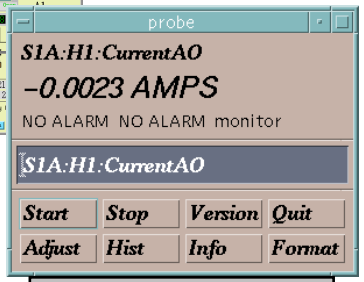
- Almost any typical Distributed Control System (DCS) function, e.g.
  - Remote control & monitoring of technical equipment
  - Data conversions and filtering
  - Closed loop control, both slow and fast
  - Access security
  - Equipment operation constraints
  - Alarm detection, reporting and logging
  - Data trending, archiving, retrieval and plotting
  - Automatic sequencing of operations
  - Mode and facility configuration control (save/restore)
  - Modeling and simulation
  - Data acquisition including image data
  - Data analysis

How does it do it?









Channel Access Client

Channel Access Client

Channel Access Client

Channel Access Client

Channel Access Server

Process Variables:

S1A:H1:CurrentAO

S1:P1:x

S1:P1:y

S1:G1:vacuum

Computer Interface

Computer Interface

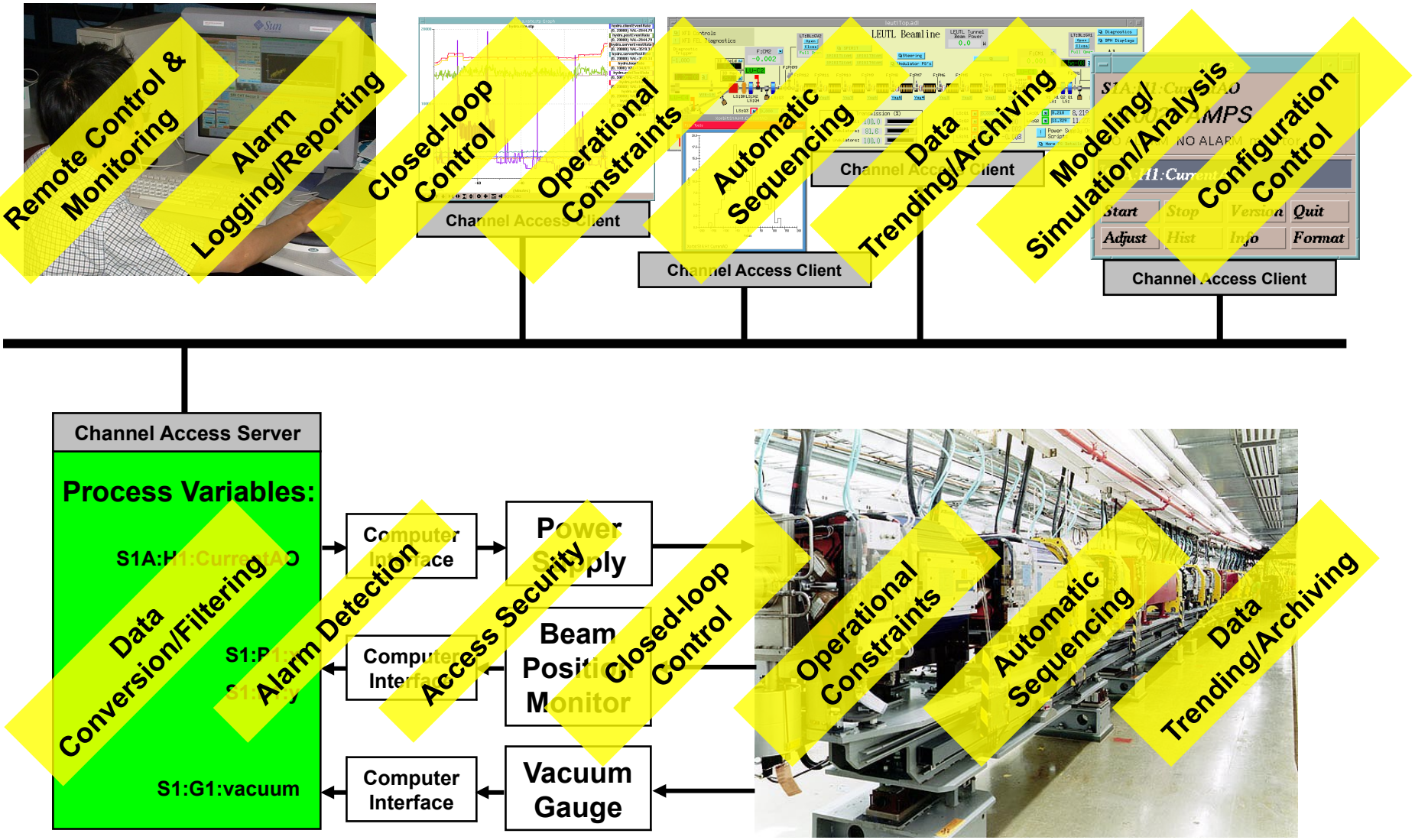
Computer Interface

Power Supply

Beam Position Monitor

Vacuum Gauge

Where are these functions performed?



The EPICS Community

Community Organization

- Leadership is informal
  - Bob Dalesio, Controls Group Leader for NSLS-2 at BNL
    - Original author of GTACS and EPICS
    - Community development
    - Organizes collaboration meetings
  - Andrew Johnson, Computer Scientist at APS, Argonne
    - Manages website, mailing lists etc.
    - Maintenance lead for EPICS Base (V3)
- Various development working groups have their own leadership
  - EPICS V4 Working Group (Greg White, Engineering Physicist at SLAC, and Andrew Johnson)
  - Control System Studio (Eric Berryman, FRIB at MSU)
  - Distributed Information Services for Control Systems (Vasu Vuppala, FRIB at MSU)
  - AreaDetector (Mark Rivers, UChicago)

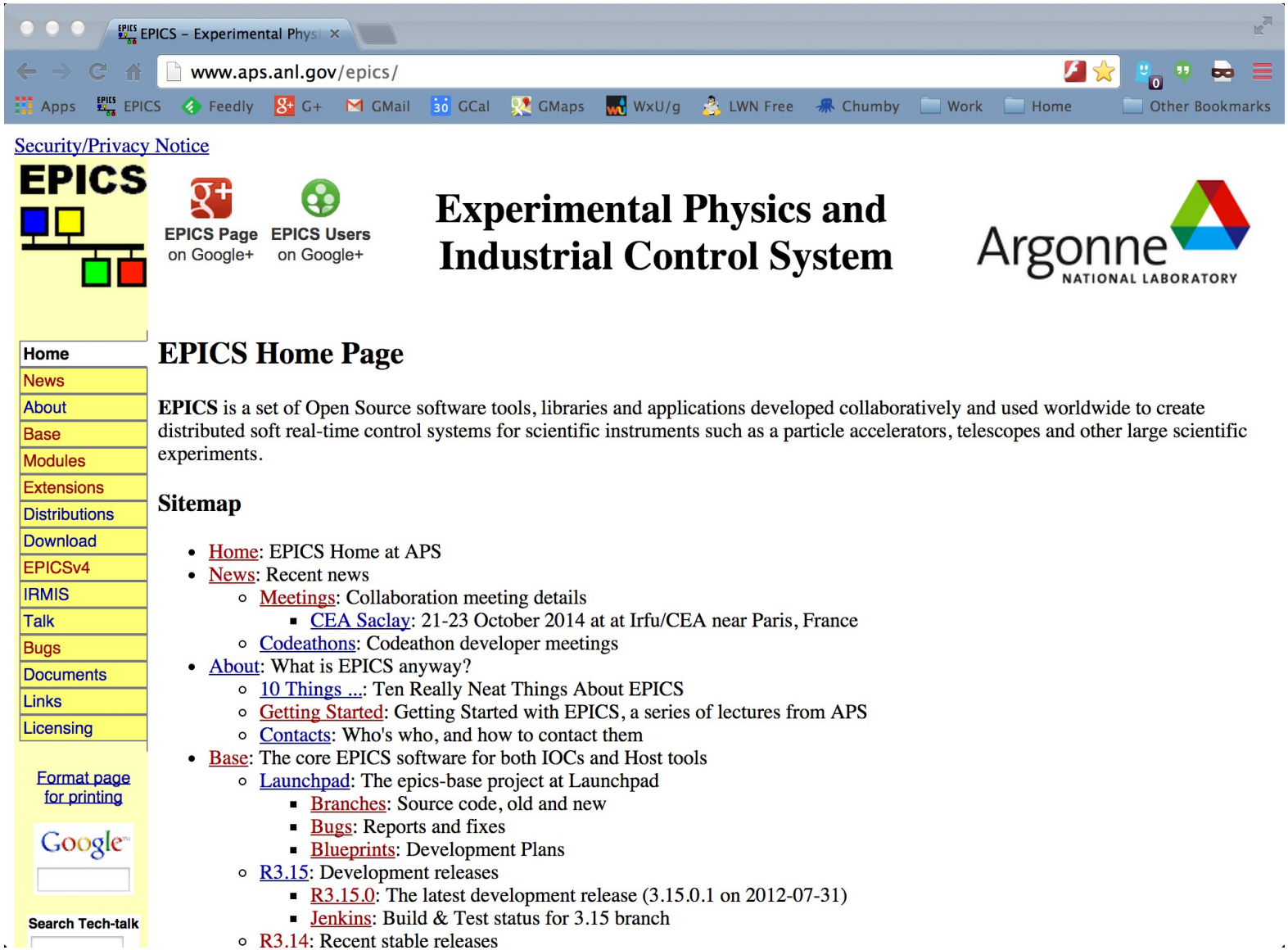
Tech-talk Mailing List

- Public mailing list (was: ‘mail exploder’)
- Free to join
  - 700+ subscribers
- Posts from non-members moderated to discourage spam
  - High signal/noise ratio
- GNU Mailman handles subscriptions, moderation, daily digest
- Archives go back to 1994
  - 2500+ messages/year in 2012 and 2013
  - RSS Feed of message subjects
- Questions and discussions occur at all levels
  - Members are usually very helpful

EPICS Website and Wiki

- Website http://www.aps.anl.gov/epics/
  - Flat-files, using PHP for templates
  - Created in 2000 to provide a central directory of EPICS content
    - Many links to other sites under Extensions, Modules, and Distributions
  - Database of supported hardware modules
    - 500+ entries
  - Only APS staff can edit website pages
- Wiki https://wiki-ext.aps.anl.gov/epics/index.php
  - Uses MediaWiki (Wikipedia) and MySQL
  - Used for miscellaneous FAQs and How-To documents
  - Accounts available for external users

EPICS Website



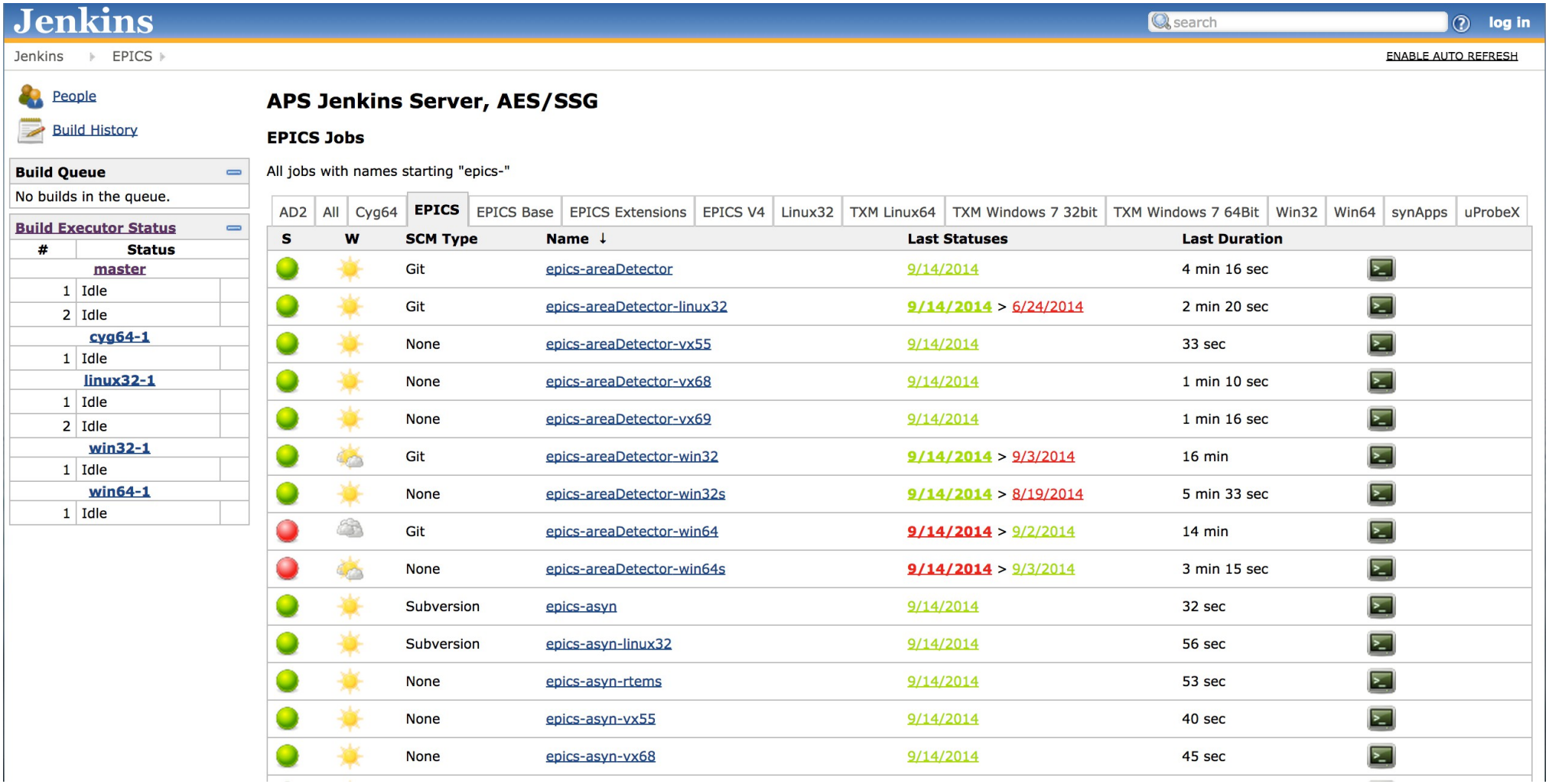
EPICS Version Control & Continuous Integration

- Early EPICS Base used SCCS for version control
- Imported code into CVS in October 1990, all SCCS history was lost
- Converted CVS history to Bazaar in December 2009
- Launchpad.net now provides branch hosting, code-reviews & bug-tracker

- EPICS Extensions and support modules from APS are still managed in CVS
  - A cvsweb installation publishes commit histories
  - Nightly snapshots of the CVS head publishes the latest code version of each module

- A Jenkins-CI server at APS tests newly committed code
  - Build slaves provided for 32- & 64-bit Linux, 32- and 64-bit Windows
  - Jenkins also used for EPICS V4 development, Asyn and AreaDetector

EPICS Jobs on APS Jenkins



Other Source Code Hosts

- APS Subversion Server
  - The synApps project is hosted here
- Several EPICS projects live on SourceForge
  - EPICS V4 development (pvdata-devel)
    - Uses Mercurial for version control
  - EPICS Applications – collection of various projects
    - Both Subversion and CVS are used
- Github is also becoming popular
  - AreaDetector, EPICS Debian, pyepics, etc. are hosted there

The APS EPICS Ecosystems

APS Accelerator Ecosystem: IOCs

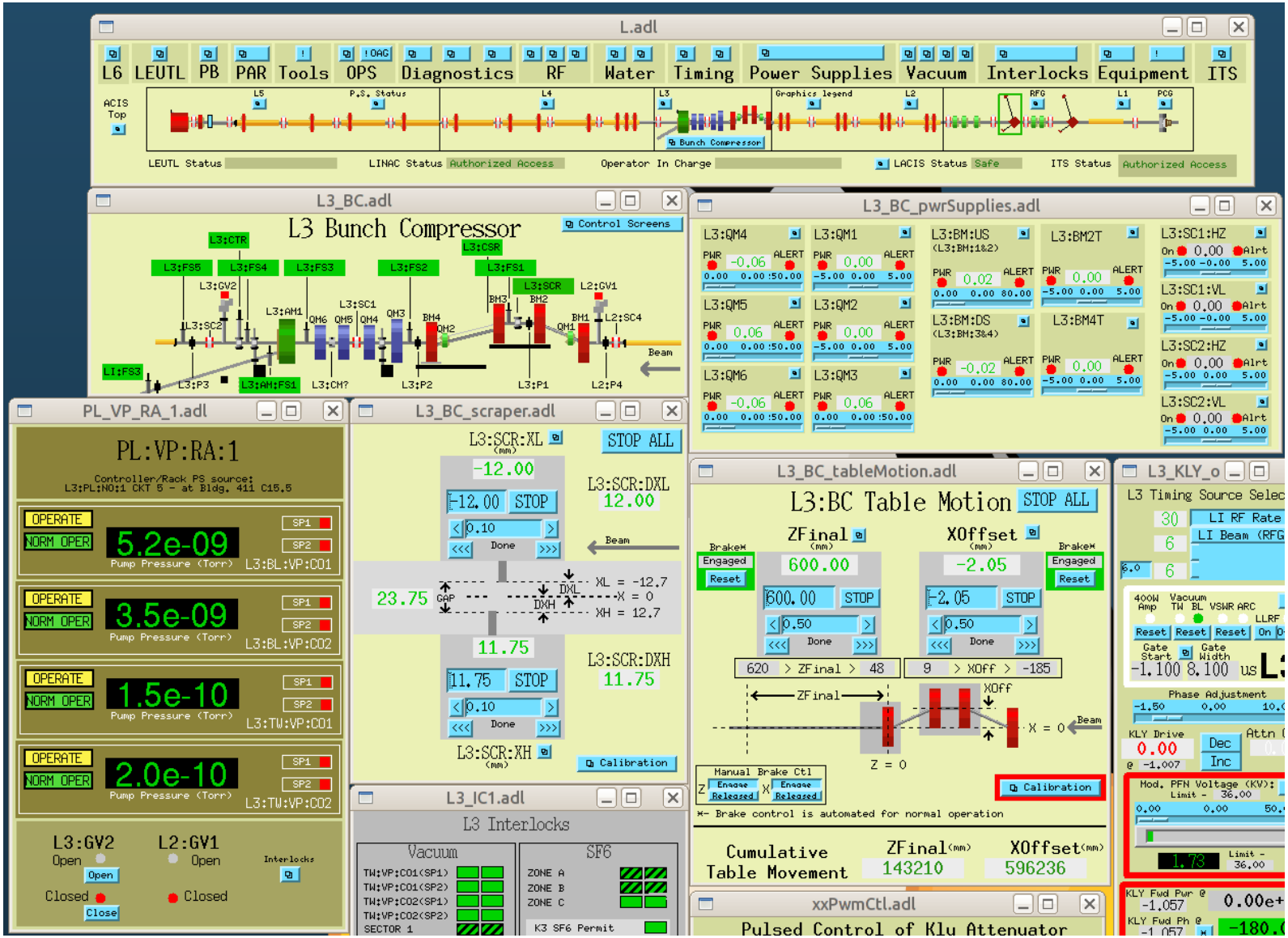
- VME-based IOCs (68K & PowerPC) running EPICS 3.13.10 – 3.14.12.3 on VxWorks
- uC-DIMM based IOCs (68K) running EPICS 3.14.11 on RTEMS
- Soft IOCs (Intel x86) running EPICS 3.14.8.2 – 3.14.12.3 on Linux (RHEL 6.5)
- Some soft IOCs (Intel x86) on Windows

- Large collection of record types, drivers & device support for all the hardware in use
- All IOCs publish standard identification and status PVs
- IOCs update a list of their PV names on a file-server for system tools to use
- APS Controls engineers and operators can connect to most IOC consoles through dedicated terminal servers for diagnostic purposes and remote reset/reboot

APS Accelerator Ecosystem: Clients

- Control room workstations run Linux (RHEL 6.5)
- Most GUI screens use MEDM, StripTool for signal plotting
- Alarm monitoring through ALH, operators develop alarm configurations
- Scientific and high-level software based on SDDS and Tcl/Tk
  - Data is archived by SDDS Data Loggers
  - Reservation system 'runcontrol' ensures that higher-level control applications cannot send competing commands to individual subsystems
- Numbered program installation system permits fast roll-back of application program upgrades
- PV Gateways give scientists and engineers read-only access to live data from outside the accelerator subnet, preventing modification of any PV values

MEDM displays for the APS Linac



APS Beamline Ecosystem: BCDA IOCs

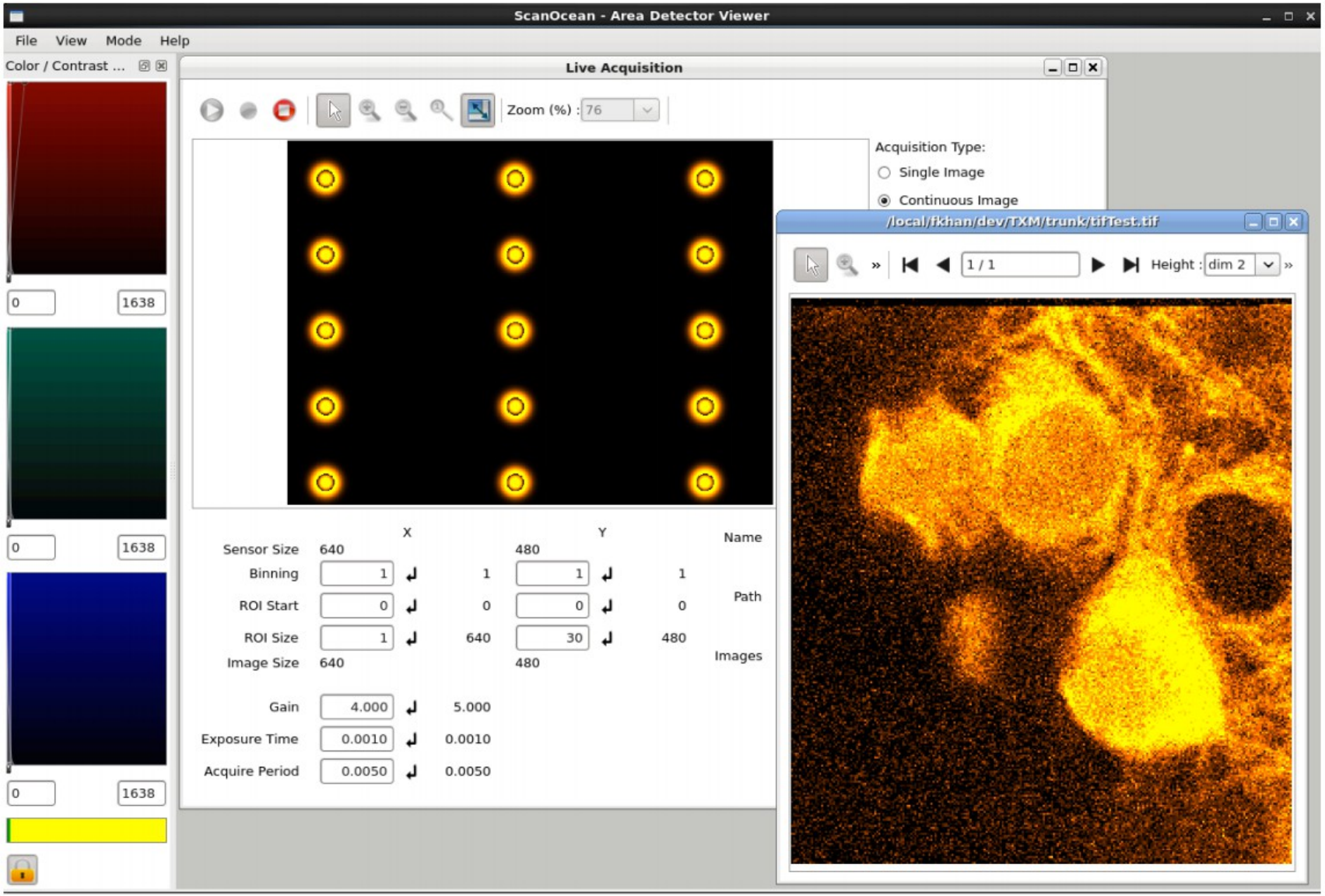
- VME-based IOCs (PowerPC) running EPICS 3.14.12.3 on VxWorks 6.9
- PC104 & PC104+ IOCs (Intel x86) running EPICS 3.14.8.2 on Vector Linux 5.1
- Soft IOCs (Intel x86) running on Linux for areaDetector

- Record types, drivers, device support and database templates provided by synApps
- Some IOCs publish standard identification and status Pvs
- alive record...
- BCDA engineers and beamline staff can connect to most IOC consoles through dedicated terminal servers for diagnostic purposes and remote reset/reboot

APS Beamline Ecosystem: BCDA Clients

- Most Beamline workstations run Linux (RHEL 6.5) with some Windows PCs
- Most GUI screens use MEDM, some use of Control System Studio
- Many users run SPEC, TXM, TomoUI, ScanSee or various custom Python and IDL programs to take scientific data
- PV Gateways give beamline IOCs and users access to live APS Accelerator data from within the beamline subnet, limiting access to PVs relevant to each beamline

TXM/ScanOcean screenshot

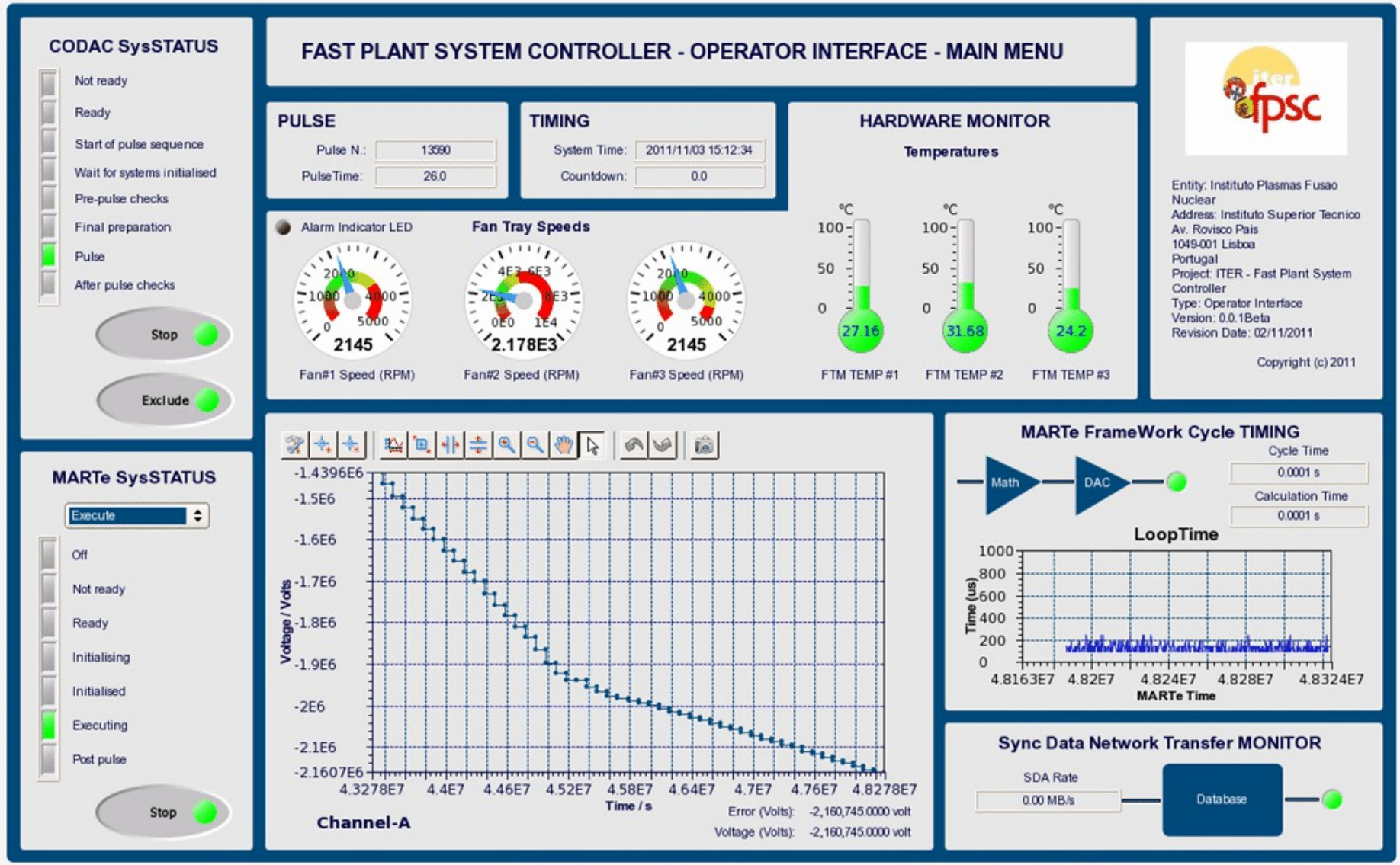


Alternative Ecosystem Choices

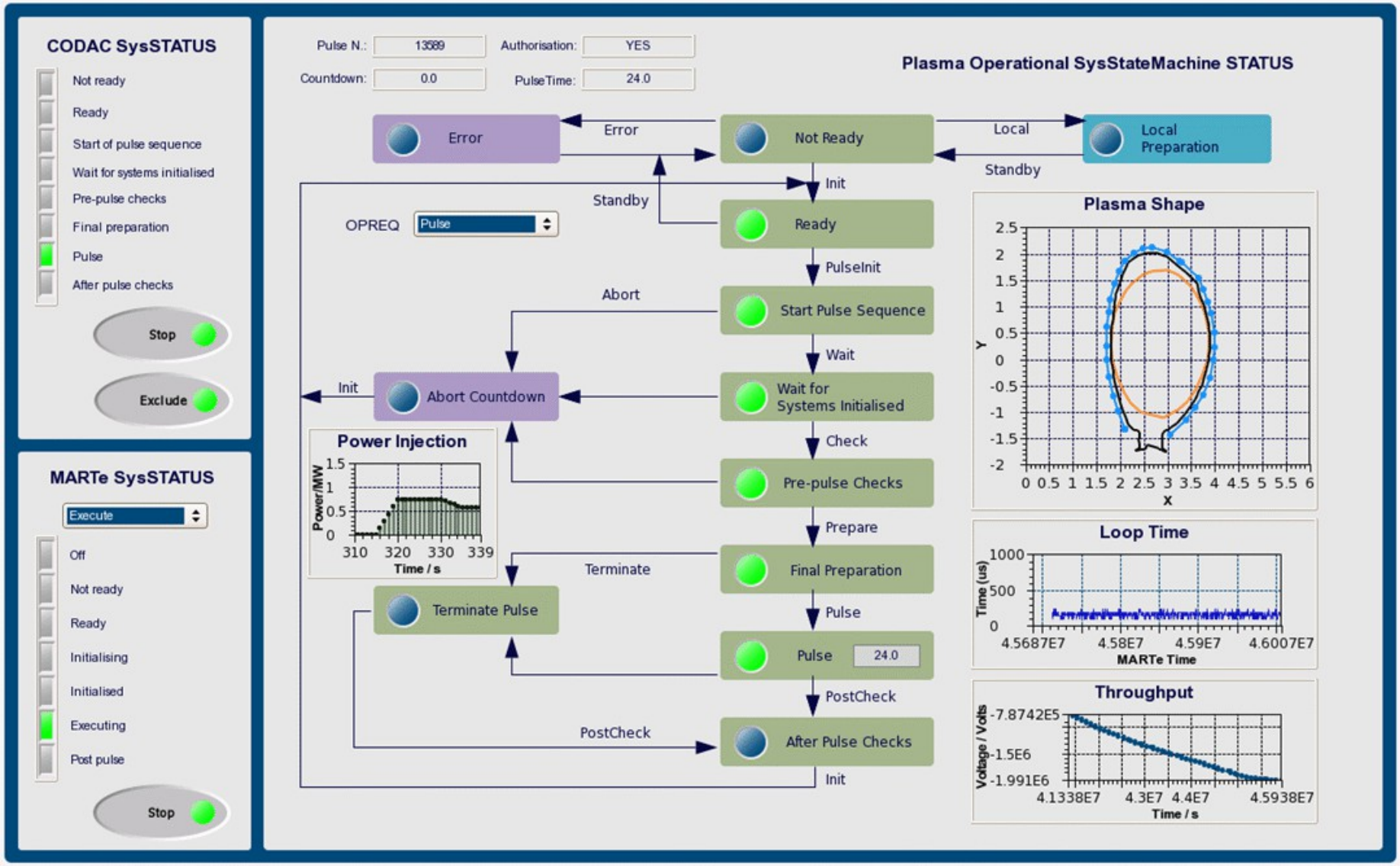
Control System Studio

- Eclipse RCP application, provides a highly integrated set of client-side tools
  - BOY: WYSIWYG editor and runtime for display screens
  - Data Browser: Plot live and archived signal data over time
  - Probe and PV Table: Interact with individual or groups of Pvs
  - PV Autocomplete: Look up PV names used previously
    - Extensions to use a PV Database or the Channel Finder
  - BEAST: Alarm monitoring and acknowledgement
  - Channel Archiver / RDB Archiver / Archive Appliance: Control data archiving
  - Logbooks: Record information in permanent electronic log with attachments
  - Scan Server: Experiment scanning with data collection triggers

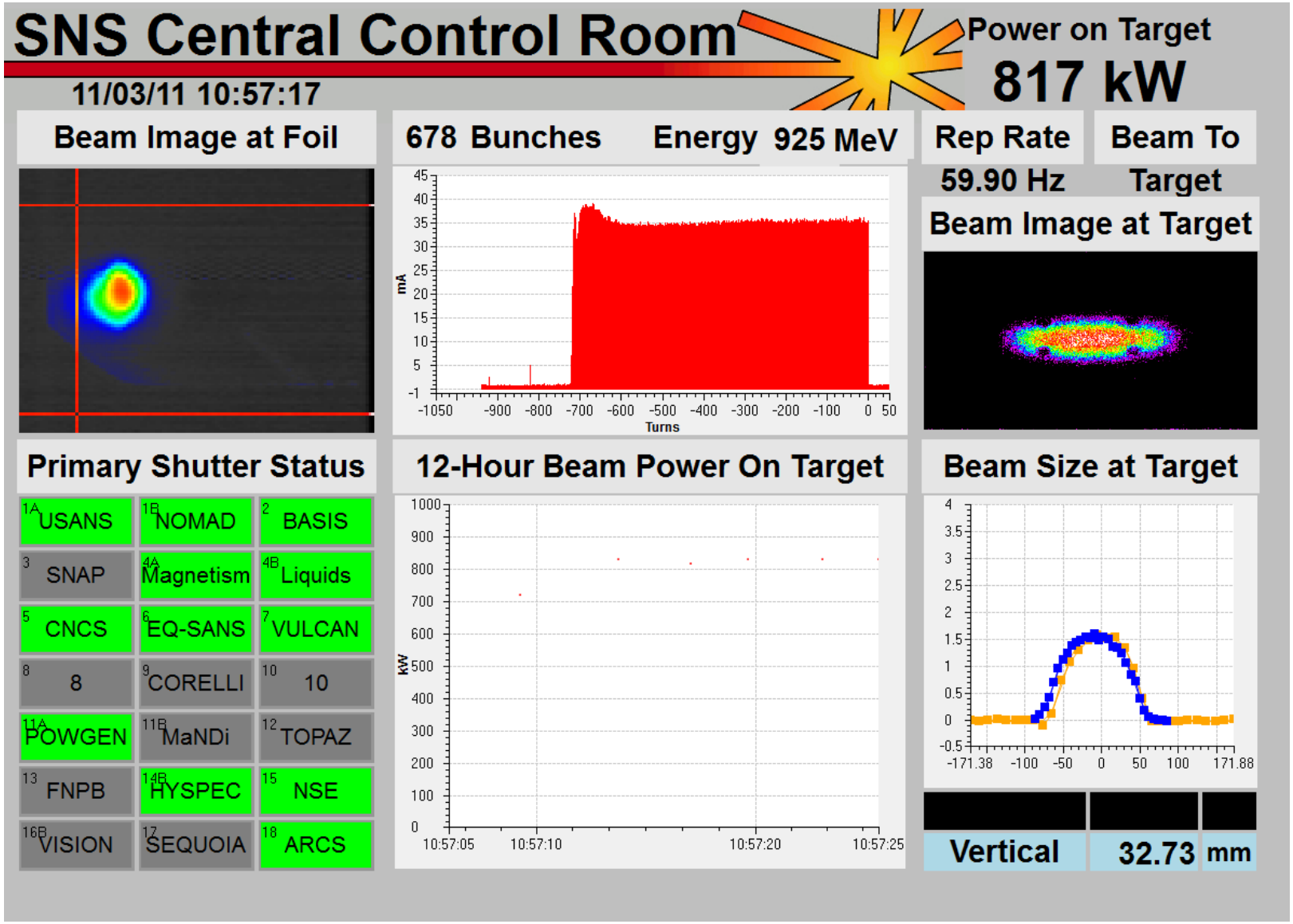
CSS-BOY Screenshots: ITER



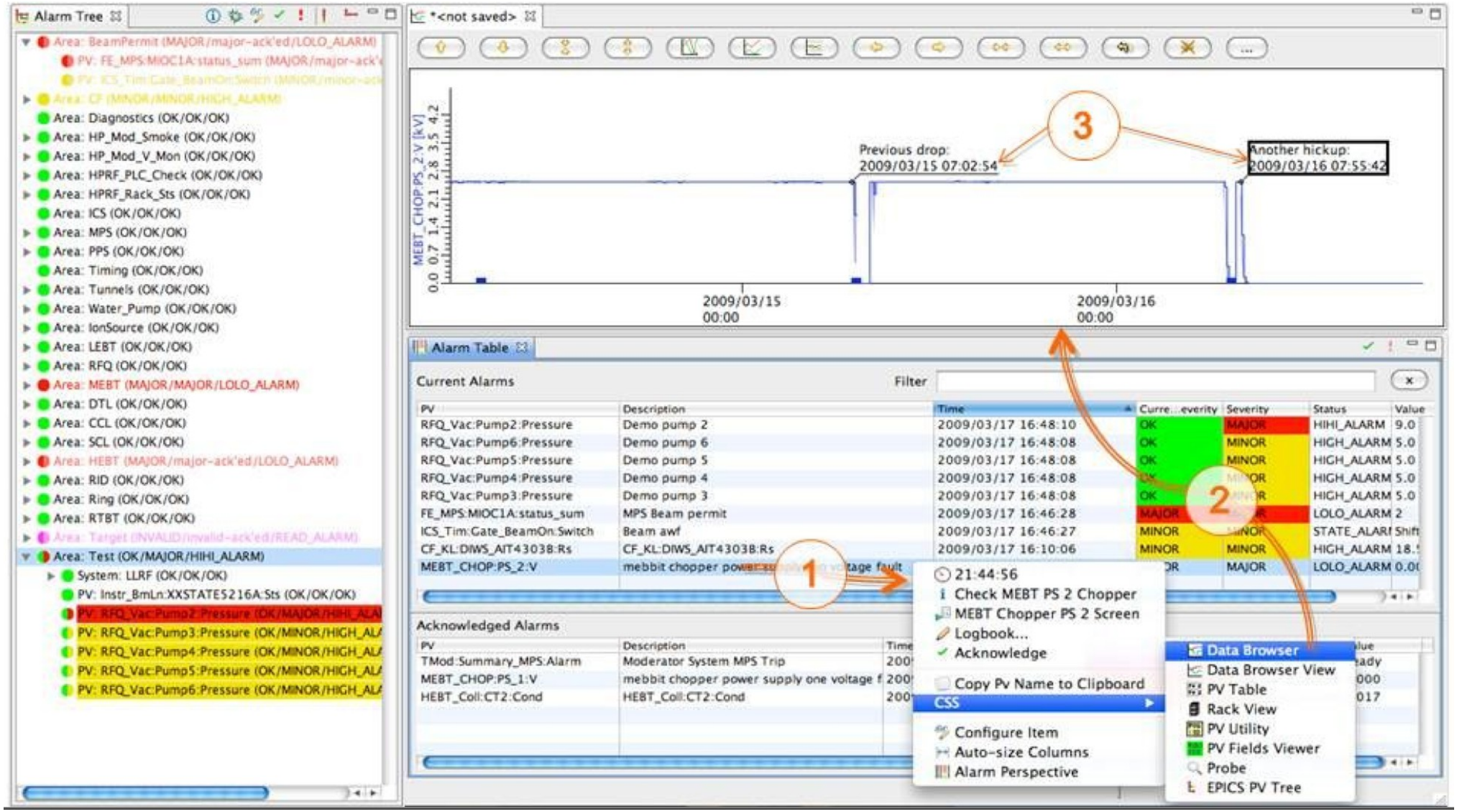
CSS-BOY Screenshots: ITER



CSS-BOY Screenshots: SNS



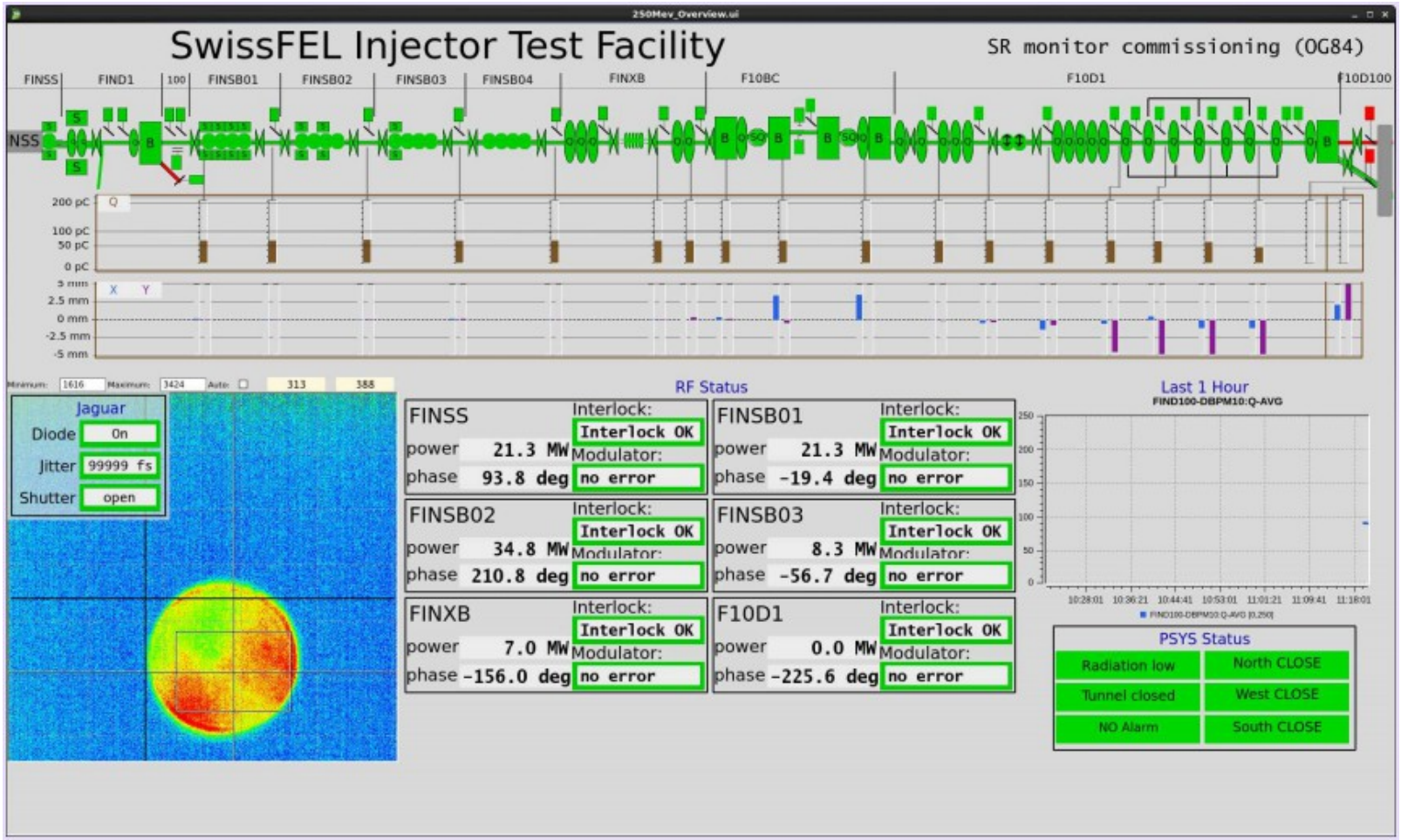
CSS-BEAST Screenshot: SNS (Annotated)



EPICS Qt

- Initiative with SBIR funding to develop a CSS-like framework for building EPICS client tools using the cross-platform Qt toolkit
- Based on epicsqt from Andrew Rhyder (Australian Synchrotron)
  - Also includes the MEDM display conversion tools and additional functionality from caQtDM by Anton Mezger at PSI
  - Integration with Python is planned
- Qt is widely used by industry and runs on a wider set of platforms than EPICS

Screenshot from caQtDM



Other Common Tools

- Data Archiving
  - EPICS data archiving tools have gone through several iterations
    - AR and ARR – 1990's
    - Channel Archiver v1, v2 – higher performance, index for retrieval
    - MYA, CZAR – starting to use RDB for index and then data too
    - BEAUTY – partitioned RDB for index and data
    - Hyperarchiver – RDB for configuration, hypertable for index and data
    - Archive Appliance – scalable, zero oversight
- Network interconnectivity
  - The EPICS PV Gateway is widely used to connect subnets, but has some limitations
  - The EpicsSharp Gateway provides higher performance with large array data
    - Written in C#, designed to run on Windows, does not use standard CA libraries

EPICS Version 4

- The V4 working group is working to replace Channel Access with a new protocol
- pvAccess
  - Efficiently transport any kind of structured data, not just value + attributes
  - Retain CA's high performance and publish/subscribe nature
  - Add capabilities for remote procedure calls (command/response)
  - Define type standards for structured data types that meet common needs
  - Provide implementations in both C++ and Java
- Marty Kraimer and Bob Dalesio (original EPICS developers) are both involved
- The latest version of CS-Studio already supports the pvAccess protocol
- Several labs are working on this
  - NSLS-2 is using it for communication between high level applications
  - SLAC is developing physics applications for LCLS-2
  - ITER and Diamond are also providing development effort