EPICS Turns 30

Bob Dalesio, Kay Kasemir, Andrew Johnson

Creation of EPICS

- The original toolset was developed under Star Wars and was named the Ground Test Accelerator Control System (GTACS)
- Collaboration began between the Argonne and Los Alamos National Laboratories in 1989
  - That Summer Marty Kraimer from Argonne worked at Los Alamos for 6 weeks learning GTACS and using it to create a magnet measurement test stand
  - The icaleps89 conference in October included several papers about GTACS
  - Controls group representatives met at Argonne in December, and agreed to collaborate on the IOC and networking software

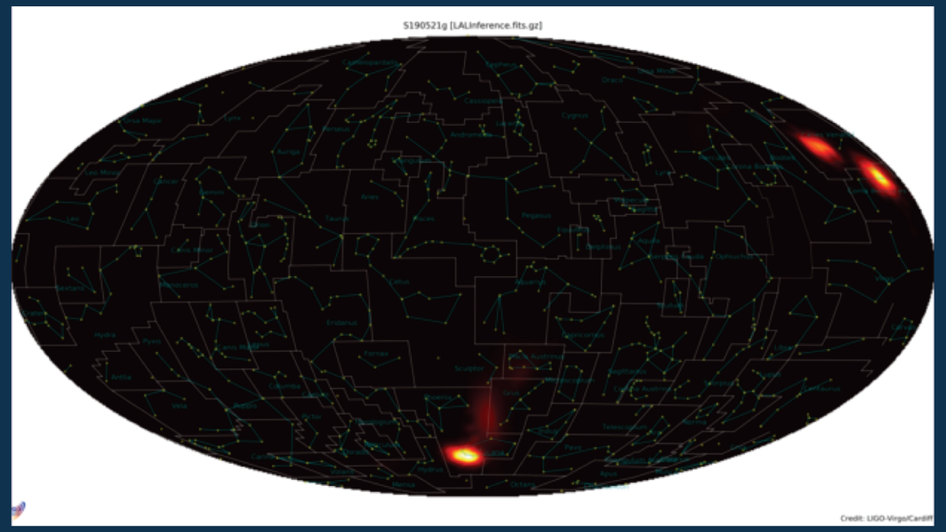
EPICS

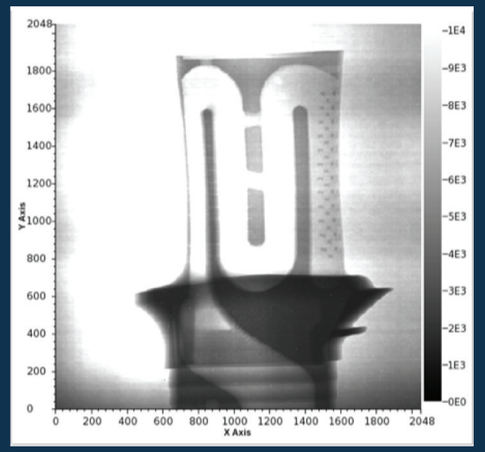
Early EPICS Users

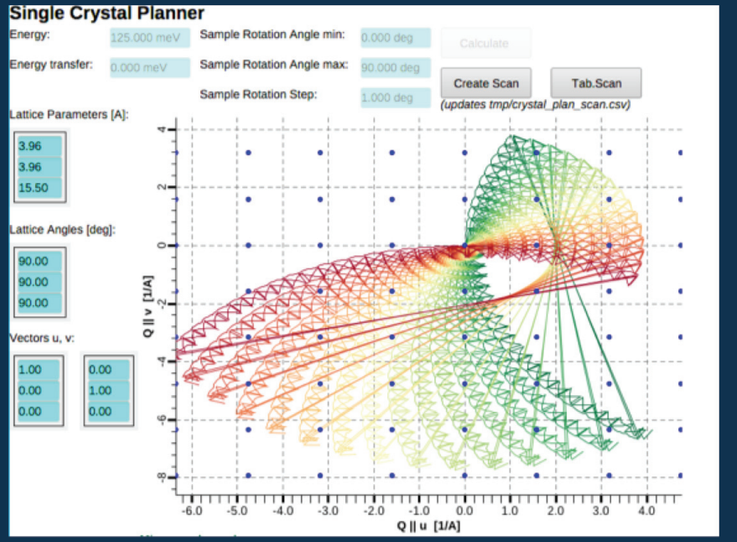
- Initially a signed license agreement was needed to join the collaboration and use EPICS; 100+ licensees
- Since 2004 the code has been freely available to anyone to use under an open source license

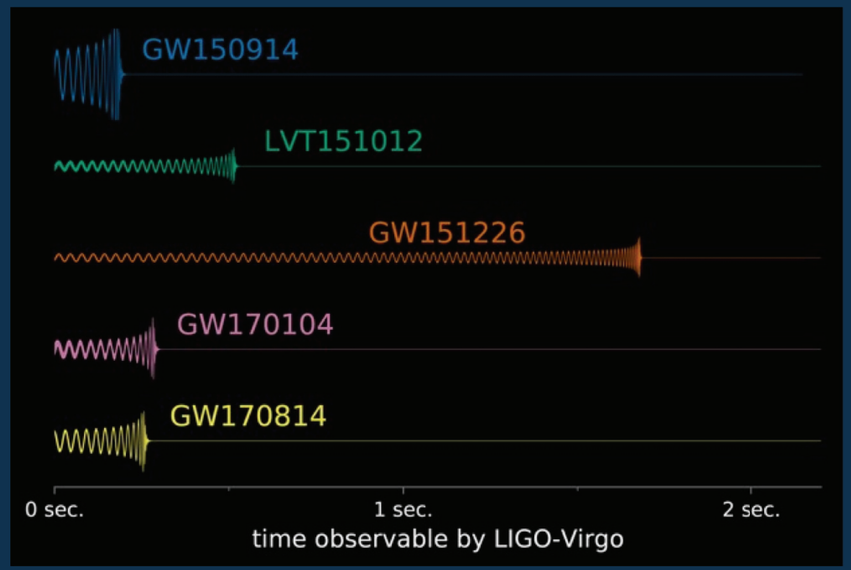
We have given up trying to track who is using EPICS.
Its success is based on loose collaboration of volunteers.

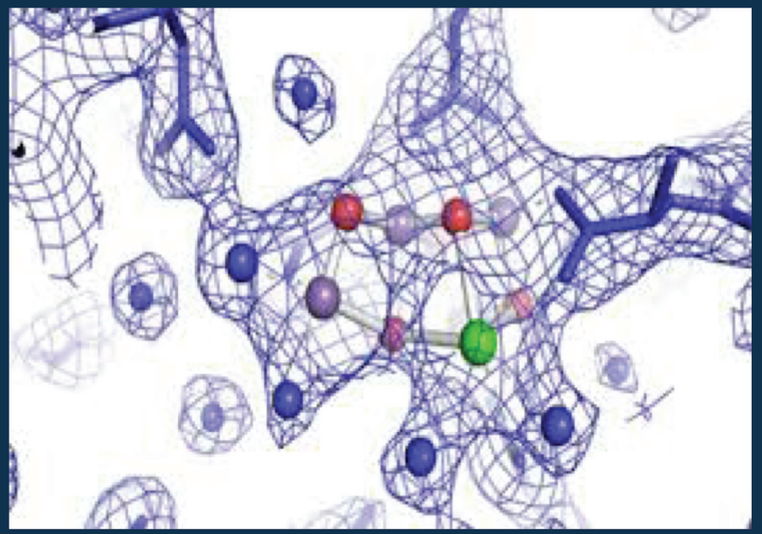
Technology Enables Discovery











Motivation Behind the Development of EPICS

- Independent Subsystem Development
- Reduce programming costs by 85%
- High Performance
- Expandable / Flexible

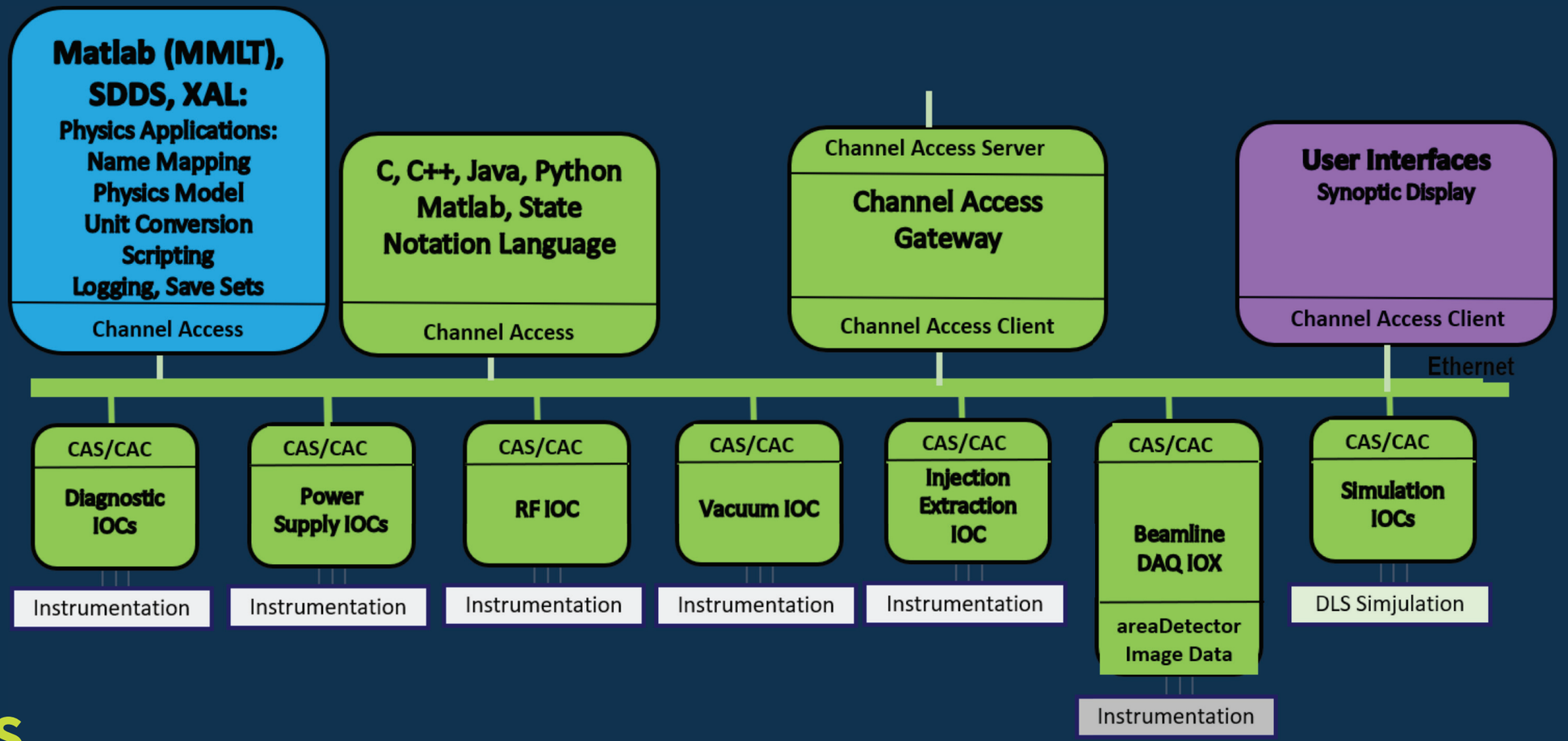
EPICS

Three Major Aspects of EPICS

- Architecture
- Tool Kit
- Collaboration

EPICS

Architecture Meets Goals and Provides Interfaces for Scientific Extensions



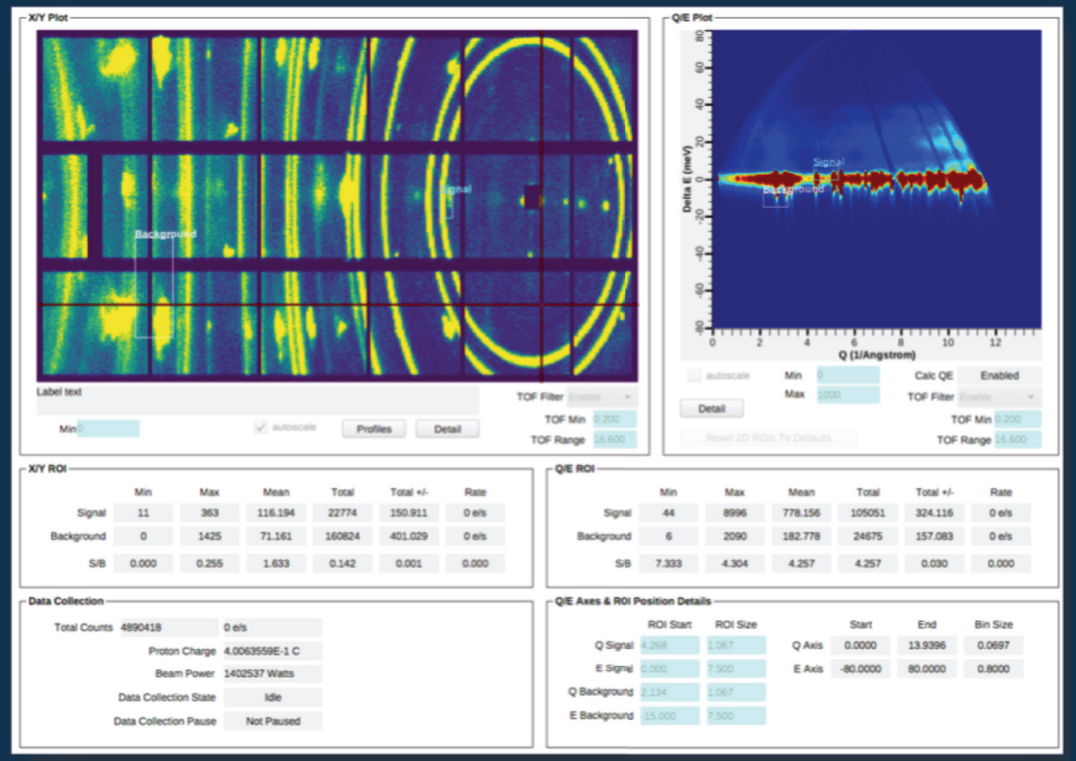
EPICS

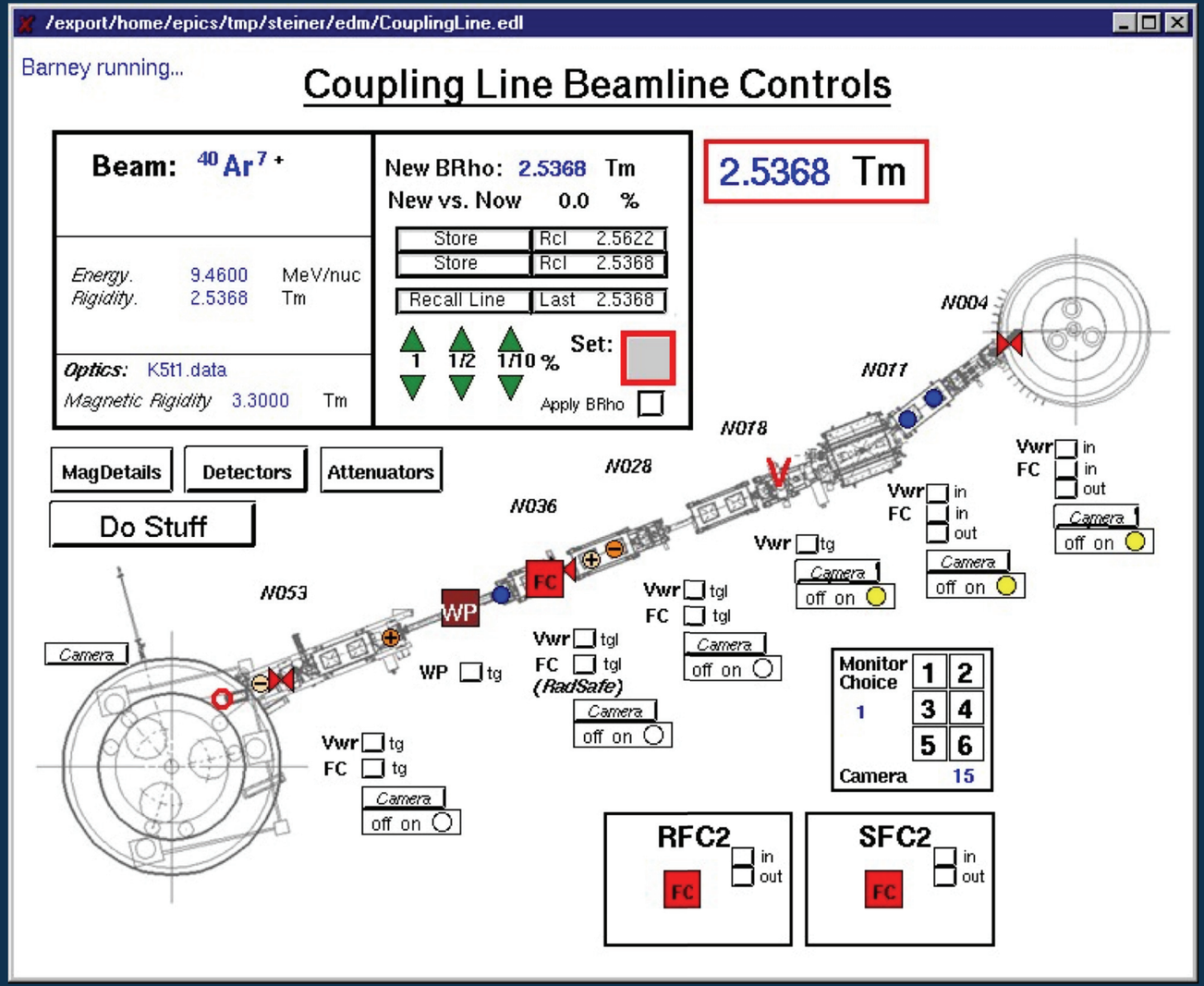
Applying and Surviving Technology

- OS: Sun OS/Solaris, VMS, Linux, VxWorks, Windows, RTEMS, macOS, iOS, Android
- IO Buses: GPIB, CAMAC, VME, VXI, RS-232, RS-485, CANbus, USB, ATCA, µTCA, PCI, cPCI, PC/104, PCIe, Ethernet, I^2C, cRIO, Control Logix, IndustryPack
- Languages: C, C++, Java, Perl, Python/Jython, C#, PHP, IDL, Tcl
- Gateways: SLC, Labview, Modbus, OPC, OPC-UA
- Display Technology: NEWS, OpenWin, Motif, X Windows, Tcl/Tk, PythonTK, Qt, SWING, Eclipse/RCP/SWT, JavaFX

EPICS

Toolkit Supports SCADA for Scientific Applications





EPICS

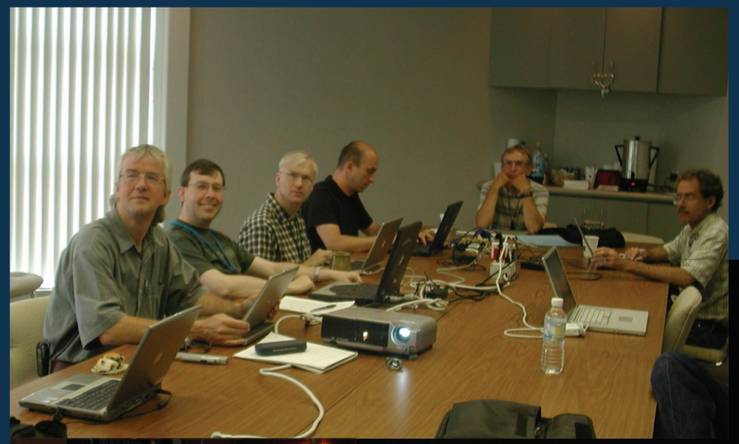
Collaboration Surviving Without A “Model”

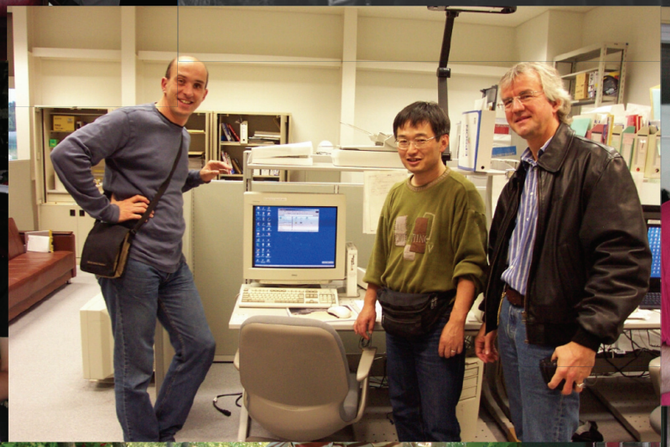
Like its architecture, the EPICS collaboration is loosely coupled

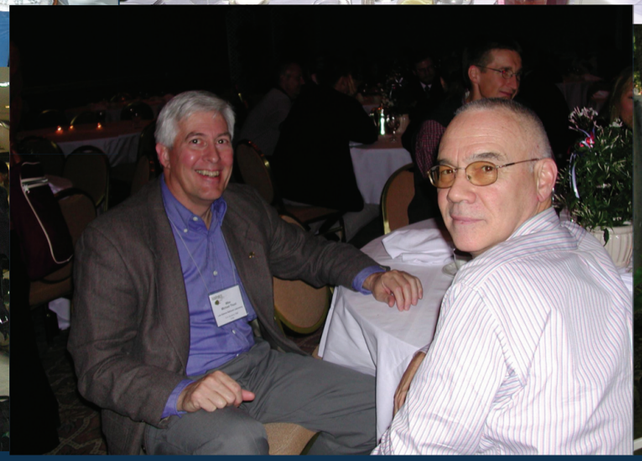
Many individuals, working diligently to support *their* community. Primary reward is the comradery of their peers, and an awareness of the science that benefits from their efforts.

EPICS

Collaboration!

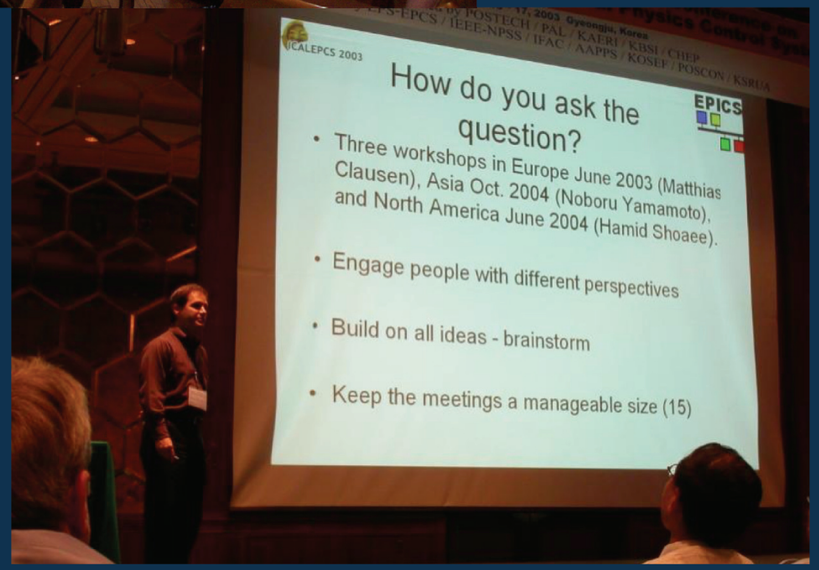






Developer Meetings

EPICS 2010 Brain Storming in 2003-2004



- What will EPICS look like in 2010?
- It was clear that the requirements of the community were expanding
- No Russian Aging Software – no clear path forward.
- We did not want to throw away the things that worked.
- Meetings were organized to brainstorm.

EPICS

Examples of presentations and points which have been discussed

Phil Duval (DESY):

TINE: What's Different …

Large data transfer: "*they want it so give it to them*"

- limited data size in transfer due only to available local memory
- beat this:

~4.5 MBytes/second to *as many clients as you want!*

~.5 Mbyte frames @ 10 Hz as TINE multicast

Data types:

- ~40 system data types (base types + names, doublets, triplets, quads, etc.)
- user-defined structures
- property overloading

e.g. can ask for "ORBIT" as float, FLTINT, FLTINTINT, etc.

can ask for piece of "ORBIT" or entire "ORBIT"

can specify starting BPM, etc.

Lessons learned
(Nick Rees)

Why was EPICS successful?

- Requirements were well understood
  - Initial implementers had done similar systems before
- Modular design
  - Individual components could be upgraded incrementally
  - Individuals didn't need to know all about everything
- Limited scope
  - Did not try and solve everything in one go.
- Discovered there were other similar applications
  - Astronomy was probably not on the agenda initially
- People and the collaboration
  - The social factors worked (and we need Bob's bad jokes!)

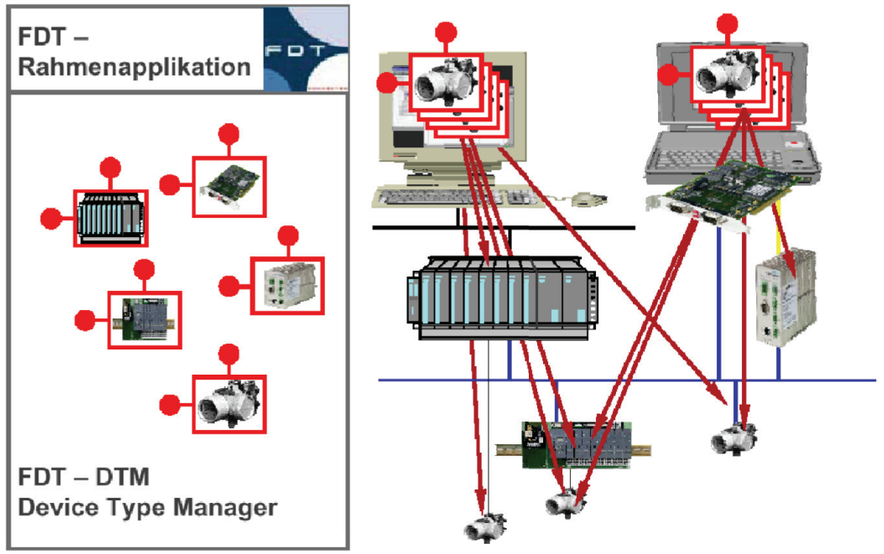
Beware

- Solving problems we don't understand well
  - If we are going to solve the high level problems in a similar way, the skill-set of the people participating must change.
  - We won't solve them in a generic way the first time around.
- Don't confuse nice-to-haves with must-haves
  - Confusing our desires as programmers with our scientific requirements
- It's the ***design***, stupid!!! Not the tools
  - EPICS is technically successful because of a few key features and its modularity, not because of VxWorks and C.

Intelligent I/O (FDT)
(Matthias Clausen)

Überblick | Feldgeräte | Integration | Konzept | Historie

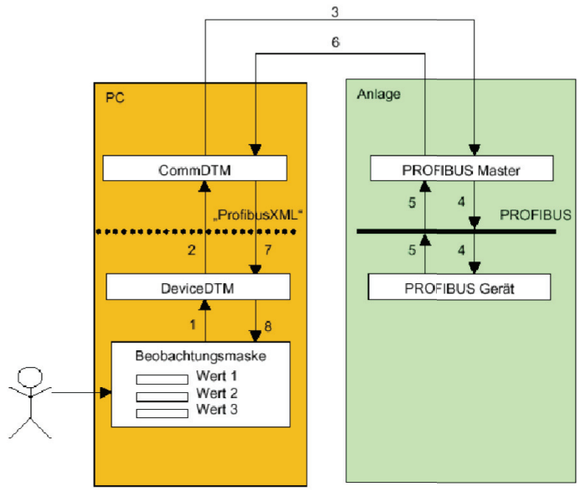
FDT - Field Device Tool – das Konzept



11.07.2002 Achim.Liebl@softing.com softing

Konzept | Rahmenapplikation | DTM | Kommunikation | Basistechnologie

Einfache Kommunikation



11.7.2002 wolfgang.fischer@softing.com softing

Results:
Interest groups

- (1) Application Services
  - Name-Service
  - Logging-Service
  - Archiving-Service
  - Alarm-Service
  - …..
  - Application Framework

Results:
Interest groups

- (2) EPICS-OO and Networking
  - How much OO does EPICS need?
  - Do we need a second network protocol?
    - CORBA?
    - DDS?
  - If only one – which one?

Results:
Interest groups

- (3) Managing and Creation of Databases
- (4) EQS (EPICS Quick Startup)
- (5) News on the use of Ethernet and intelligent I/O (collection wiki only)

Interest groups run dedicated wiki‘s

(Public Web pages)

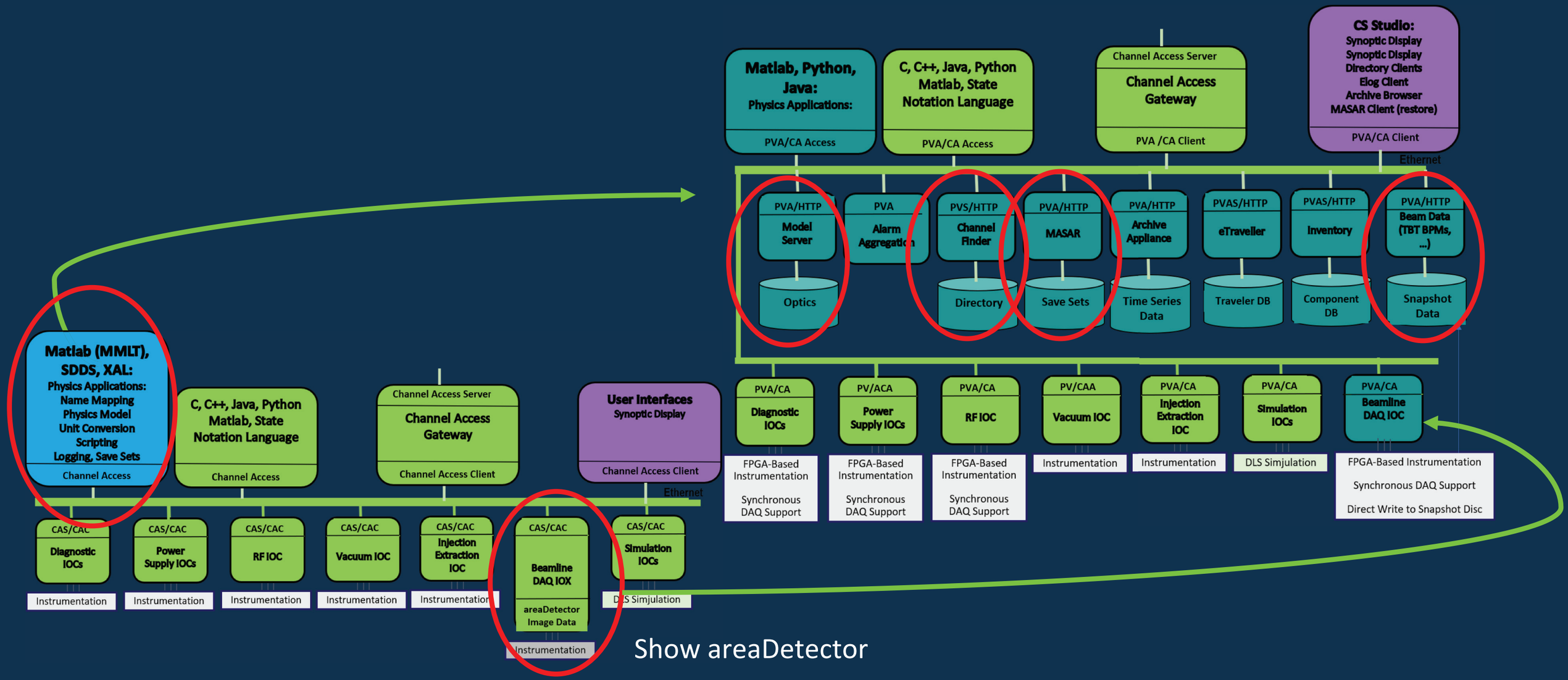
Most Interest Groups have members outside the EPICS collaboration.

Motivation Behind the Development of EPICS7

- Upwardly Compatible with previous releases
- Improve Ease to Expand Data Types
- Send Meta-Data Efficiently
- Support Middle Layer Services: Directory, SnapShot etc..
- Expand MetaData to Support High Rate Data:
  - *n*-dimensional Arrays
  - Tables
  - Statistical Samples
- Improve Time Stamp and Alarm MetaData

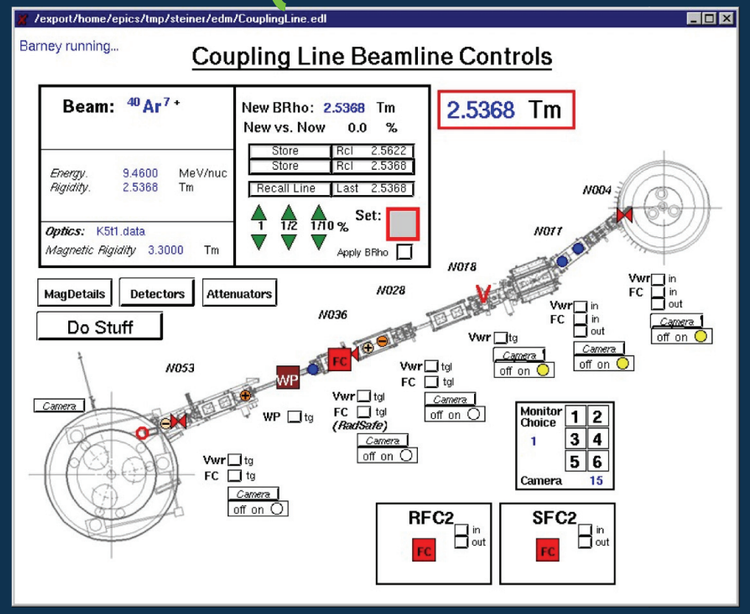
EPICS

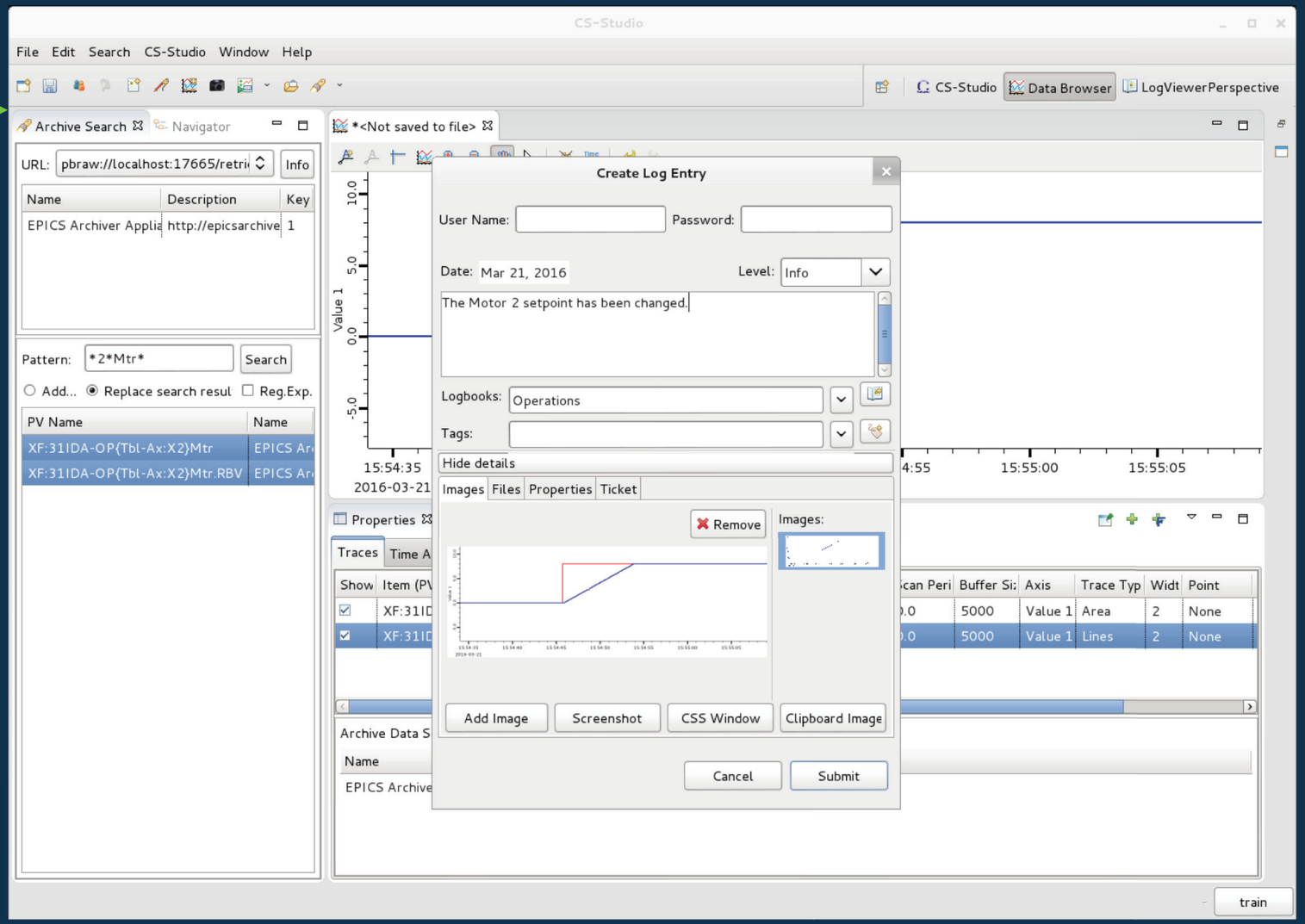
Architecture Supports FPGA and RDB Integration



Show areaDetector

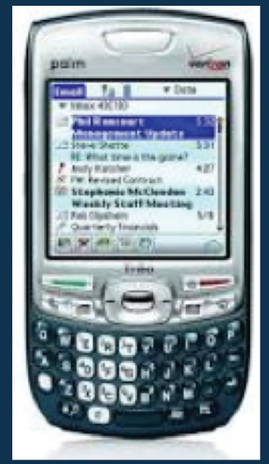
Toolkit Supports Better Workflow and Higher Rate Data with *n*-dim Arrays and Middle Layer Services





Evolution of Tools

- Underlying technologies change
  - EPICS (IOC records, Channel Access) abstracts details of bus, I/O etc.
  - Update to PVAccess will be more visible



- Graphics support evolves
  - Need new Display Manager every ~10 years
  - New tools are bi-lingual

EPICS and Time

Ahead:

- Unix 32-bit epoch ends Jan. 19, 2038, 03:14 GMT
- EPICS time-stamps will work until 2106!

Behind:

- Stuck with Release 3.14.x from 2002-2014

Accommodating:

- More than one clock? Don’t know who’s right?
- ‘General Time’ can register them all!

Collaboration Meetings 2018-2019

Developer Meetings and Codeathons

Developer Meetings
PVA Services
AreaDetector Developers
CS-Studio Developers
EPICS Core Developers
BlueSky Workshop
Open Hardware
Motion Control

EPICS – The Next 30 Years

- Significant parts of the architecture have evolved to support modern evolving technology and improve workflow.
- The toolkit is well poised to support scientific exploration in the upcoming years.
- It is hoped that the collaboration will continue to commit the resources essential to attracting a critical mass of dedicated and top quality people required to support the community (right?)

EPICS

Acknowledgements

- The entire EPICS community, for their contributions large and small

- Marty Kraimer, Bob Dalesio and Jeff Hill, who in 2009 received the first ICALEPCS Lifetime Achievement Award for “the creation, development, and nurturing of the EPICS control system toolkit and the EPICS collaboration.” http://icalepcs2009.spring8.or.jp/LAA.html

EPICS