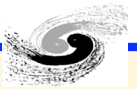
EPICS Overview

EPICS Seminar/Workshop
Institute of High Energy Physics (IHEP)
Beijing, China
March 1-5, 2010

Overview Presented by:
Kazuro Furukawa, KEK
古川 和朗

Based on Presentations by:
Ned Arnold, John Maclean, Andrew Johnson,
Kay Kasemir, Bob Dalesio, Marty Kraimer.



Information Sources

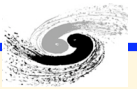
- Main WWW site
  - www.aps.anl.gov/epics/
    - Documents/Training
- This talk taken from the following training presentations
  - RRCAT2009, SSRF2008, APS, PSI, SNS, USPAS2007, USPAS2003,
  - Ned Arnold, John Maclean, Andrew Johnson, Kay Kasemir, Bob Dalesio, Marty Kraimer
  - The talk from PSI did not list authors.

What is EPICS?

- A Collaboration
- A Control System Architecture
- A Software Tool Kit

What is EPICS?

- A Collaboration
  - Began in 1989 between LANL/GTA & ANL/APS
    - (Bob Dalesio & Marty Kraimer)
  - Over 150 license agreements were signed *before* EPICS became "open source"
  - Recent EPICS collaboration meetings
    - 100+ Attendees
    - 30+ Institutions
    - 75+ Presentations
  - List server; *tech-talk*: the collaboration in action
  - Collaborative efforts vary
    - Assist in finding bugs
    - Share tools, schemes, and advice

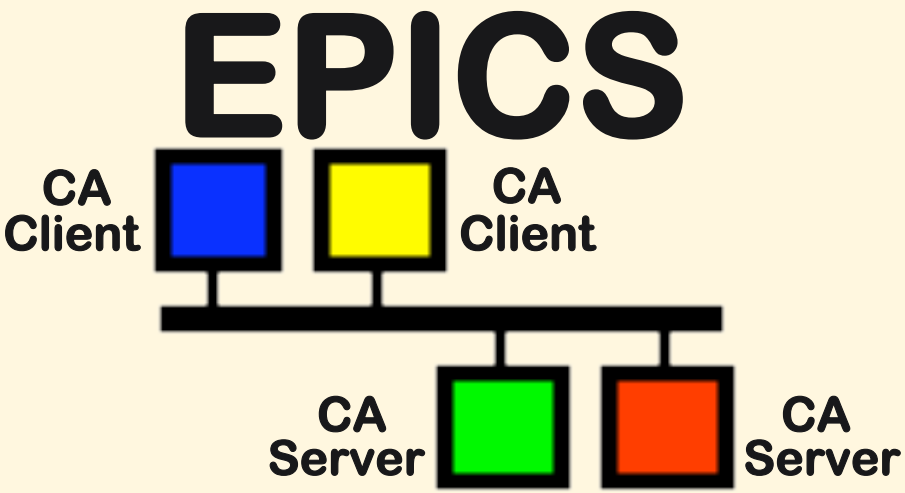


What is EPICS?

- Early Major Collaborators
  - America
    - ANL, LANL, LBL, ORNL, SLAC, JLAB
  - Europe
    - DESY, BESSY, PSI
  - Asia
    - KEK
  - Also several observatories
- Recent Major Collaborators
  - America
    - TRIUMF, CLS, BNL, Universities
  - Europe
    - DIAMOND Light Source, ITER
  - Asia/Australia
    - J-PARC, IHEP, SSRF, TLS, PAL, K-STAR, RRCAT, Australian Synchrotron
- Many Other Collaborators/Users

What is EPICS?

- A Collaboration
- A Control System Architecture
  - Network-based “client/server” model (hence the EPICS logo)

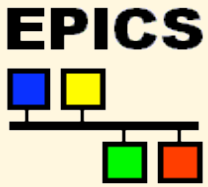


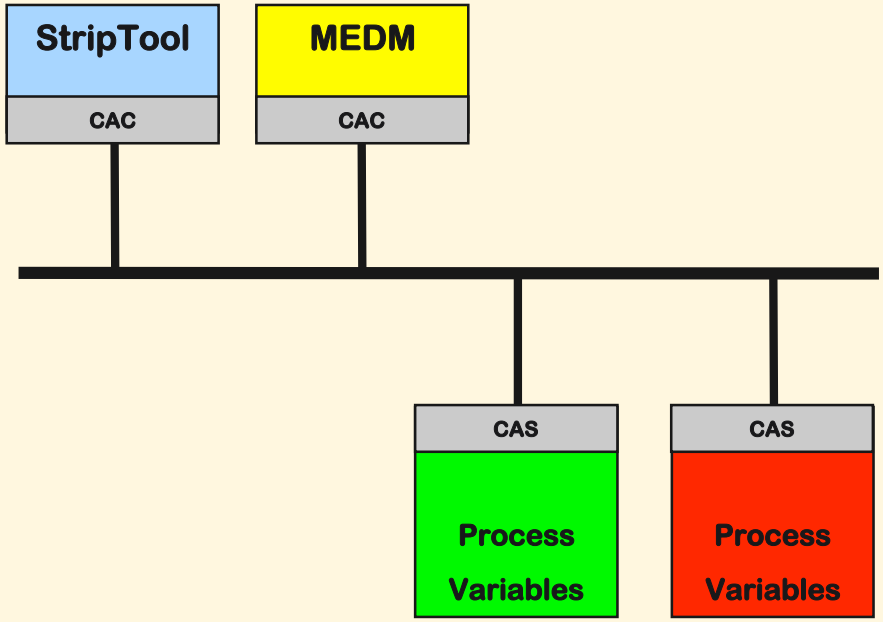
For EPICS, *client* and *server* speak of their Channel Access role

i.e. Channel Access Client & Channel Access Server

What is EPICS?

- Channel Access *clients* are programs that require access to ***Process Variables*** to carry out their purpose



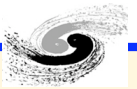


The "service" that a Channel Access *server* provides is access to a *Process Variable

* A *Process Variable* (PV) is a named piece of data.

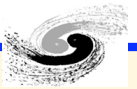
What is EPICS? Vocabulary

- EPICS
  - Experimental Physics and Industrial Control System
- Channel Access
  - The communication protocol used by EPICS
- Process Variable
  - A piece of named data referred to by its PV name
  - The primary object of the Channel Access Protocol
- Channel
  - A synonym for Process Variable
- Channel Access Server
  - Software that provides access to a Process Variable using the Channel Access Protocol
- Channel Access Client
  - Software that requests access to a Process Variable using the Channel Access Protocol



What is EPICS? Vocabulary

- IOC – Input Output Controller
  - A computer running *iocCore,* a set of EPICS routines used to define process variables and implement real-time control algorithms
  - *iocCore* uses database records to define process variables and their behavior
- Soft IOC
  - An instance of *iocCore* running as a process on a "non-dedicated" computer (i.e. a computer that is performing other functions as well)
- Record
  - The mechanism by which a Process Variable is defined in an IOC (using *iocCore*)
  - Dozens of record types exist, each with it's own attributes and processing routine that describe its functionality



What is EPICS: Vocabulary

- EPICS Base
  - IOC, 'hard' and 'soft'
  - Database (records)
  - Channel Access: Server and Client
  - Previously Driver/Device support: Now available as modules
  - Previously Sequencer: Now unbundled.
- Extensions
  - Client tool that is not part of base
  - EDM, StripTool, …
- Module
  - IOC support that is not part of base. Mostly device/driver support
  - AsynDriver, etc, etc.

What is EPICS?

- Process Variable

  - A ***Process Variable*** (PV) is a named piece of data associated with the machine (e.g. status, readback, setpoint, parameter)

  - Examples of PV names and values:

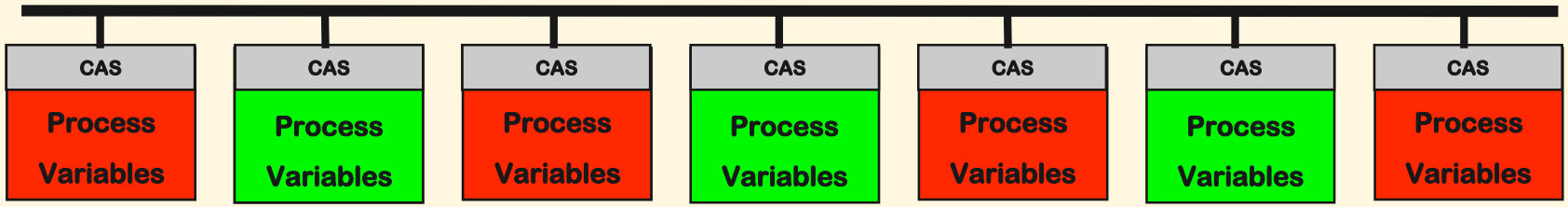
    - S1:VAC:reading 3.2e-08 torr
    - LINAC:BPM4:xPosition -0.323 mm
    - BOOSTER:gateValvePosition 'OPEN'
    - S3:DIPOLE:PS:setPoint 123.4 Amps
    - APS:Mode 'Stored Beam'
    - BL3:HISTOGRAM {3, 8, 1, 2, 56, 44, 32, 43, 3, 5, 1}

What is EPICS?

- Process Variable
  - A ***Process Variable*** is a named piece of data with a set of attributes
  - Examples of Attributes:
    - Alarm Severity (e.g. NO_ALARM, MINOR, MAJOR, INVALID)
    - Alarm Status (e.g. LOW, HI, LOLO, HIHI, READ_error)
    - Time stamp
    - Number of elements (array)
    - Normal Operating Range
    - Control Limits
    - Engineering Unit Designation (e.g. degrees, mm, MW)

What is EPICS?

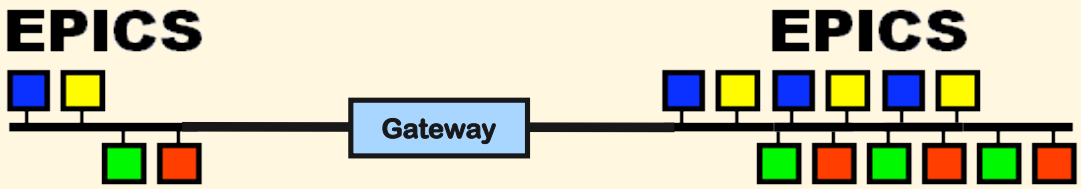
- A Control System Architecture
  - Network-based "client/server" model where the basic data element is a Process Variable
  - The Channel Access Protocol defines how Process Variable data is transferred between a server and client
  - The entire set of Process Variables establish a *Distributed Real-time Database* of machine status, information and control parameters



What is EPICS?

- By default, Channel Access traffic is constrained to a single subnet, but configuration options can direct traffic elsewhere

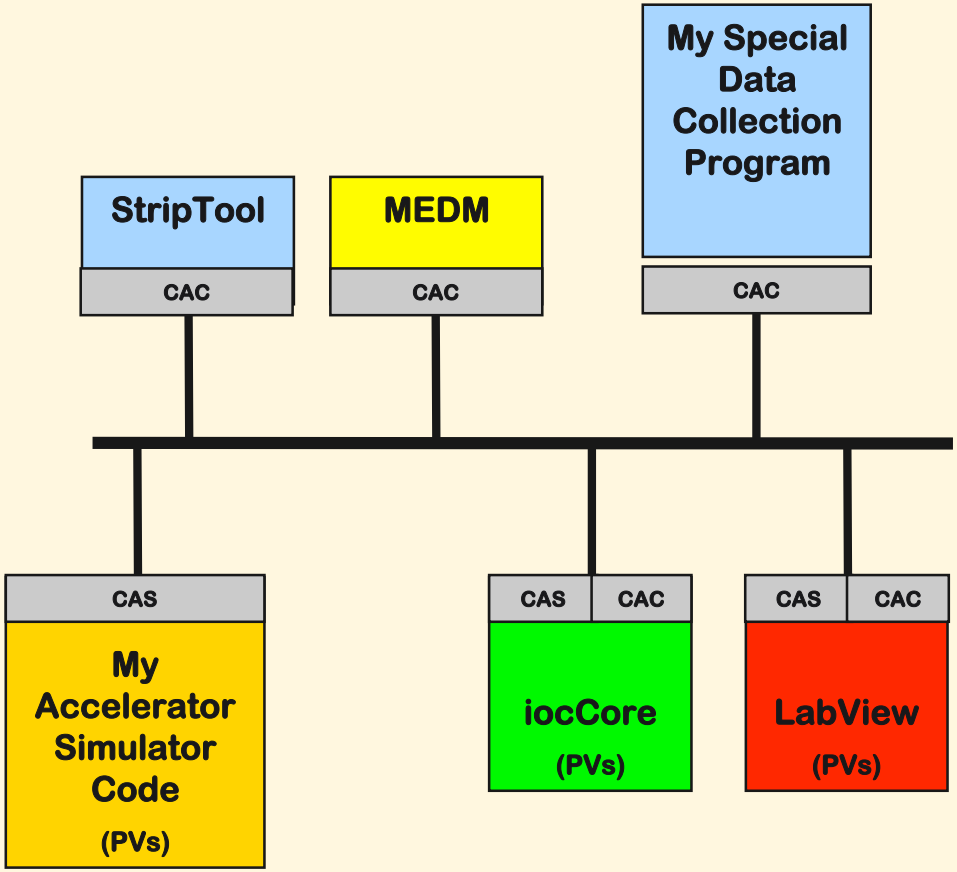
- Physical hierarchies can be implemented using switches, routers, and gateways



What is EPICS?

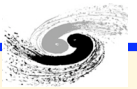
- Any tool/program/application that abides by the Channel Access protocol could be described as “EPICS Compliant”.

EPICS can be viewed as a “toolkit” of EPICS compliant programs. One can select the appropriate tool for their need or develop their own.



What is EPICS?

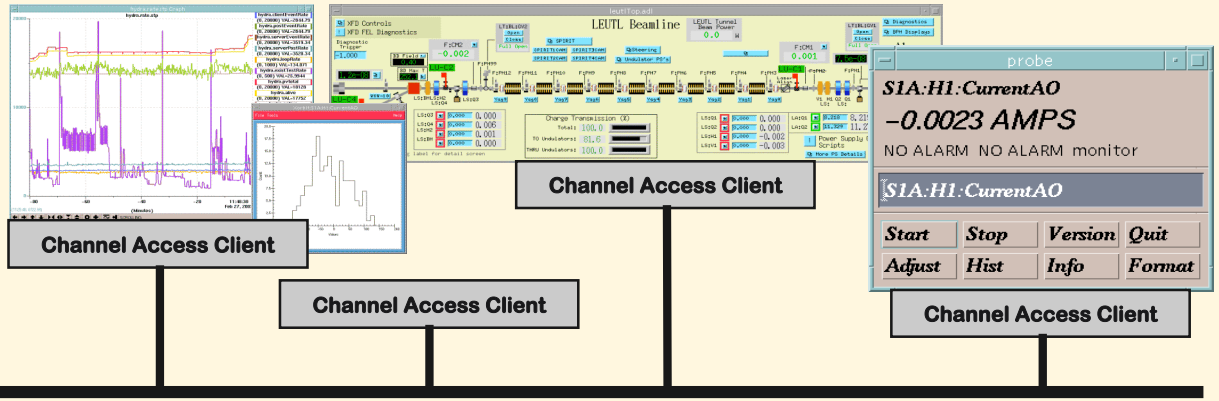
- A Collaboration
  - A world wide collaboration that shares designs, software tools, and expertise for implementing large-scale control systems
- A Control System Architecture
  - A client/server model with an efficient communication protocol (Channel Access) for passing data
  - A distributed real-time database of machine values
- A Software Toolkit
  - A collection of software tools collaboratively developed which can be integrated to provide a comprehensive and scalable control system

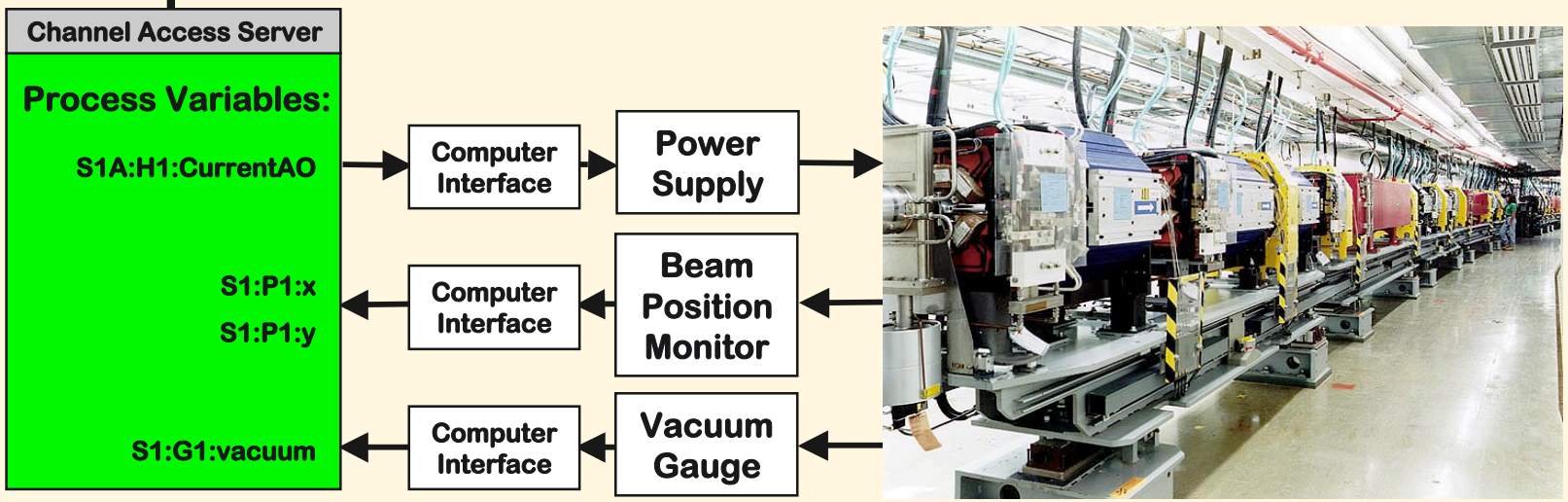


So What Does it Do?

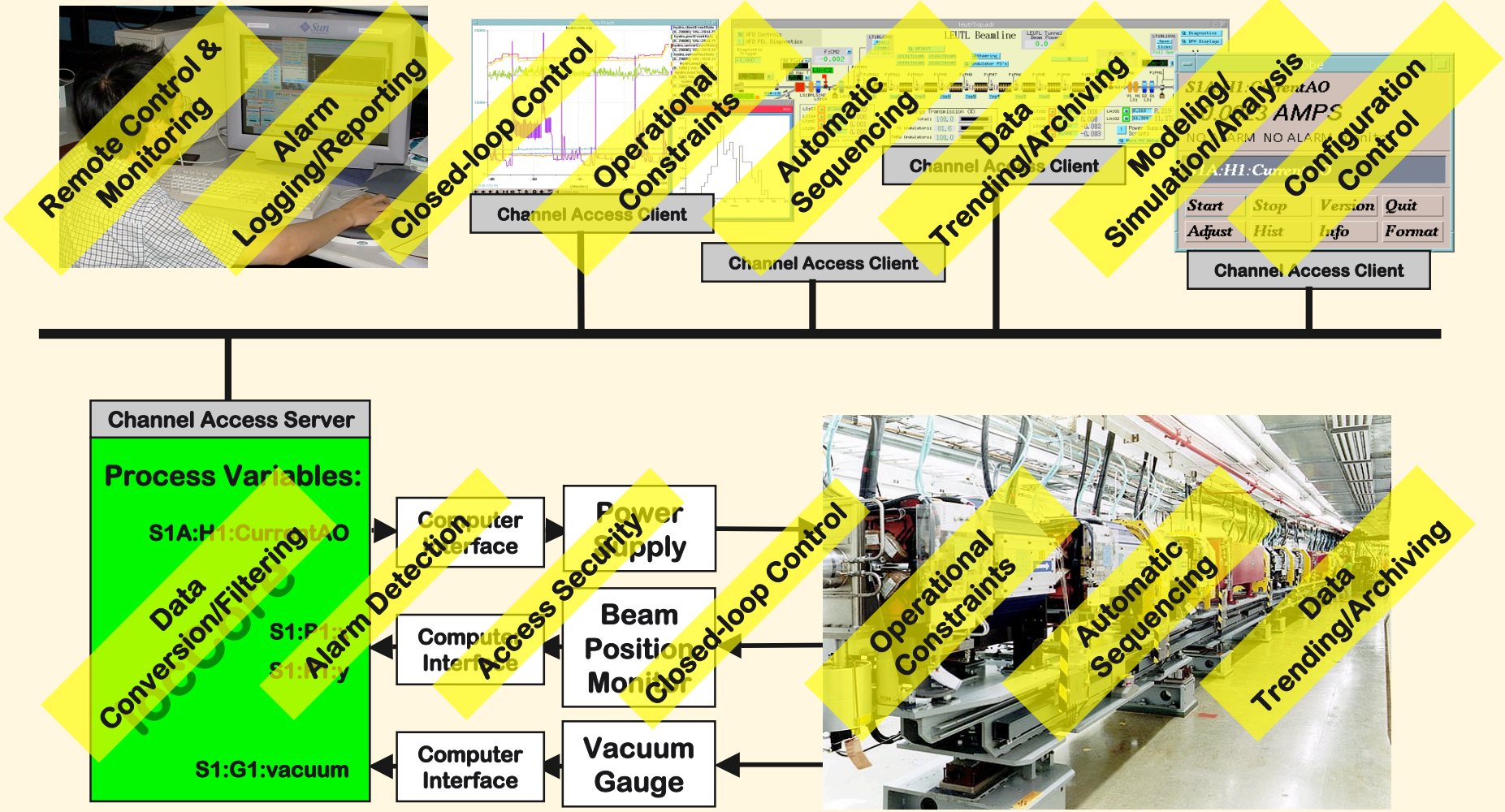
- EPICS tools are available to accomplish almost any typical Distributed Control System (DCS) functionality, such as:
  - Remote Control & Monitoring of Technical Equipment
  - Data Conversion/Filtering
  - Closed Loop Control
  - Access Security
  - Equipment Operation Constraints
  - Alarm Detection/Reporting/Logging
  - Data Trending/Archiving/Retrieval/Plotting
  - Automatic Sequencing
  - Mode & Facility Configuration Control (save/restore)
  - Modeling/Simulation
  - Data Acquisition
  - Data Analysis

How does it do it?

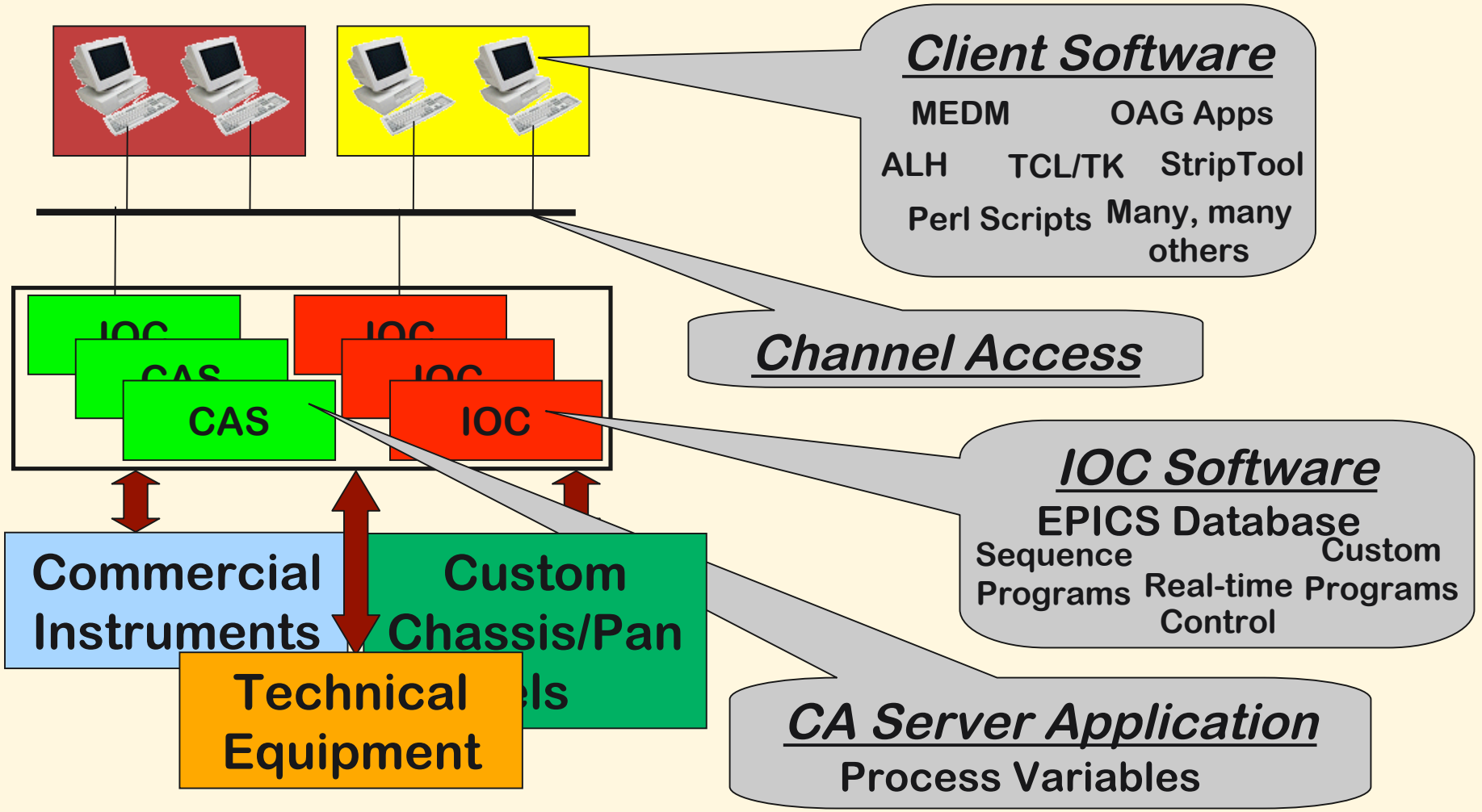




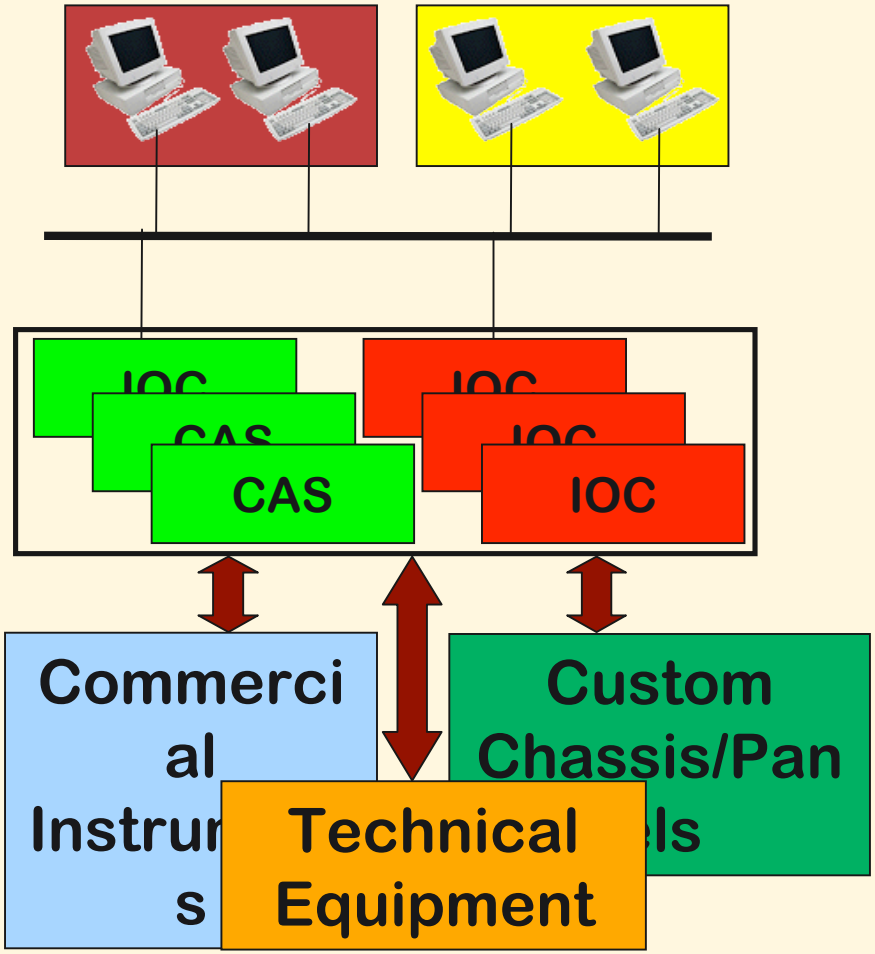
Where does it do it?

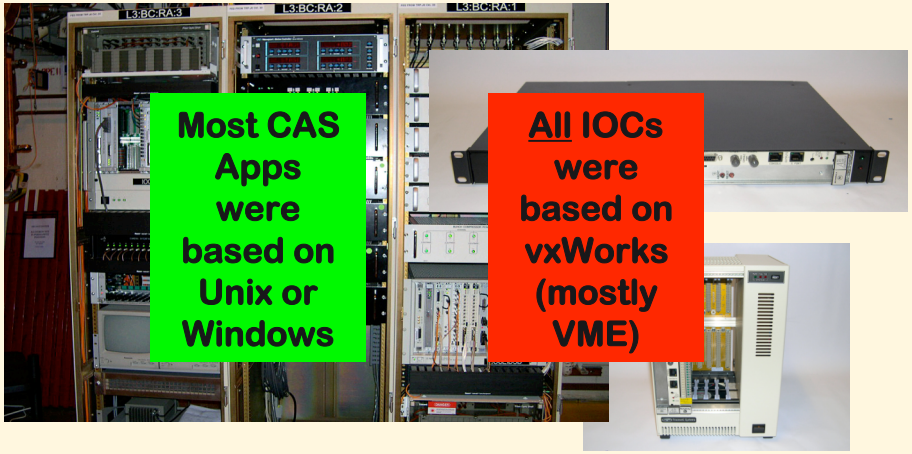


Canonical Form of an EPICS Control System

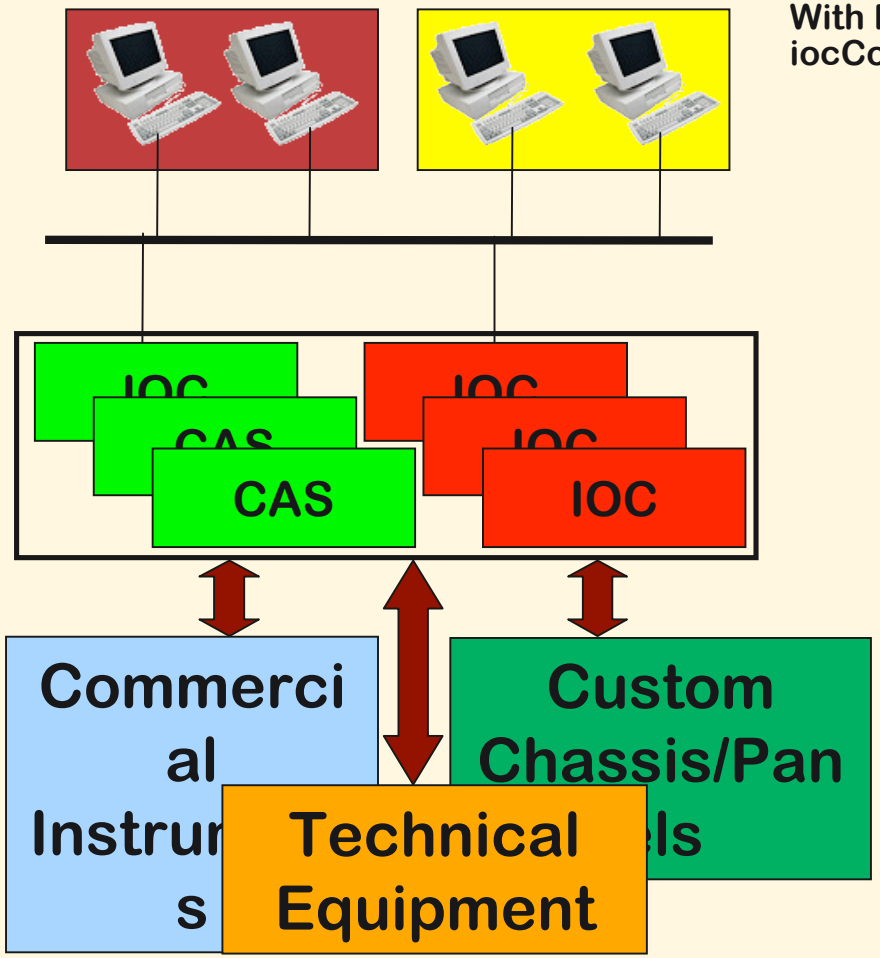


Typical Realizations of an EPICS System





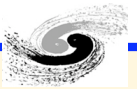
Typical Realizations of an EPICS System



With Release 3.14, the operating system limitations for iocCore have been removed.

Introducing the IOC

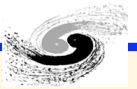
- Input Output Controller
- A computer running software called "*IOC Core*"
- The computer can be:
  - VME based, running vxWorks (only choice until Release 3.14) or RTEMS
  - PC running Windows, Linux, RTEMS
  - Apple Macintosh running OSX
  - UNIX Workstation running Solaris, HP-UX, etc.
  - (ITRON based system)
- Usually has Input and/or Output devices attached
- An EPICS control system must consist of at least one Channel Access Server (usually an IOC)
- An IOC has one or more *databases* loaded. The database tells it what to do



IOC Database

- Configuration instead of Coding
- 'iocCore' software loads and executes 'Records'

- Example Assignment:
  - Read a temperature sensor
  - Open/close a valve when value is above resp. below some threshold



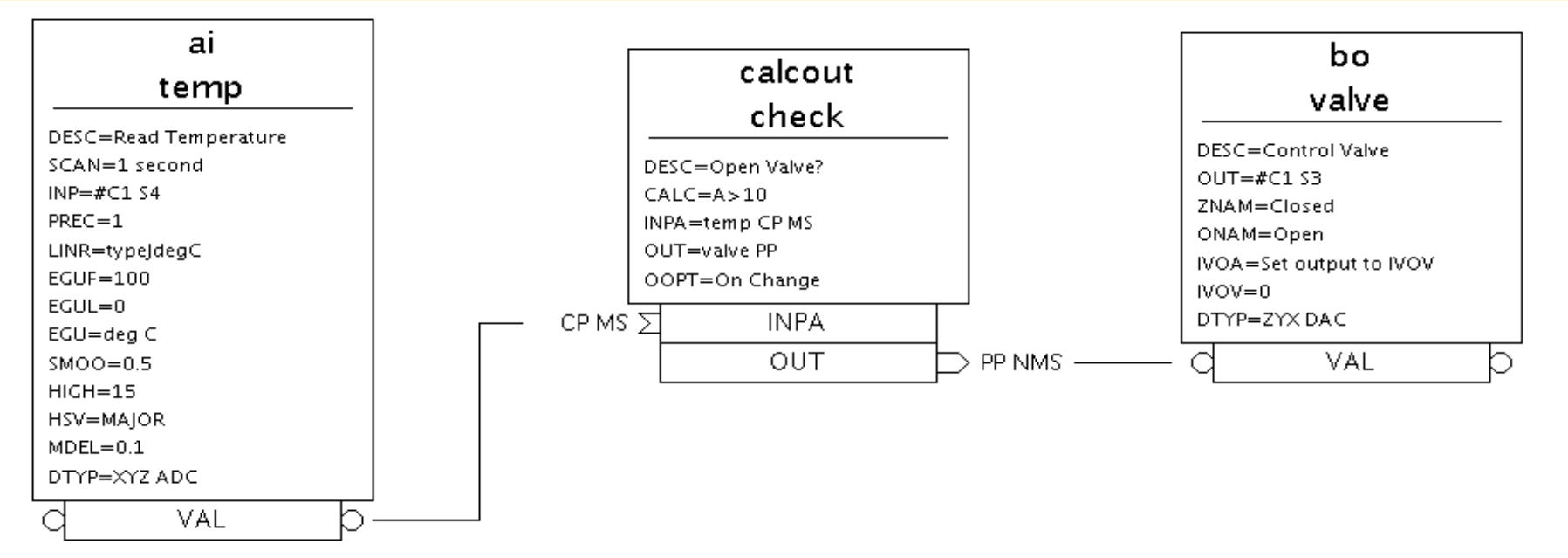
The Example in simplified Code

```
Sensor temp = open_device(…);

Valve valve = open_device(…);

Loop:

      if (temp.value() > 10)
       valve.open();
      else
       valve.close();

     delay(1.0);
```

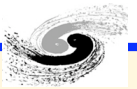
What we omitted

- Error checking
- Code comments
- Apply some smoothing to the temperature reading to filter noise.
- Send current temperature and valve state to network clients (operator display).
- Attach a time stamp to the data, so that network clients can see for example when the valve was last opened.
- Send warnings if the temperature is close to the threshold, or an alarm when way above.
- Allow runtime changes of the threshold from the remote operator interface.
- Allow runtime changes to the scan rate.
- Maybe allow runtime changes to the device address?

This IOC 'Database' does all that



- At first glance, this might look much worse than the code, but...
  - that was simplified code.
  - there's no way the full code for the above would fit on one screen.
  - after learning more about the database (~2 days), this becomes much more readable than somebody else's custom code for the same functionality.



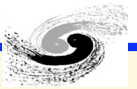
Some Detail on an EPICS 'Record'

```
record(ai, temp)
{
  field(DESC, "Read Temperature")
  field(SCAN, "1 second")
  field(DTYP, "XYZ ADC")
  field(INP,  "#C1 S4")
  field(PREC, "1")
  field(LINR, "typeJdegC")
  field(EGUF, "100")
  field(EGUL, "0")
  field(EGU,  "deg C")
  field(SMOO, "0.5")
  field(HIGH, "15")
  field(HSV,  "MAJOR")
  field(MDEL, "0.1")
}
```

- Configuration instead of Programming
- "SCAN=1 second" instead of starting periodic thread, delaying until next multiple of 1 second, locking required resources, …
- "SMOO=0.5" configures the smoothing algorithm.
- Almost any field in any record is accessible via network at runtime
  - Change scan rate, smoothing, …

IOC Database

- A single record often handles the scanning, signal conditioning, alarming of a temperature, pressure, or similar analog reading.
- Combined with binary and computational records, it can express most of the data flow logic for a front-end computer
  - Avoiding the pitfalls of real-time, multithreaded and networked programming.
- One can have thousands of records in one IOC.
  - Well suited for systems with high signal counts, like vacuum or water systems with relatively simple logic but many, many sensors and valves.
- kHz-rate processing with record chains is doable
  - Of course limited by CPU. Not 1000 of kHz rate-records…



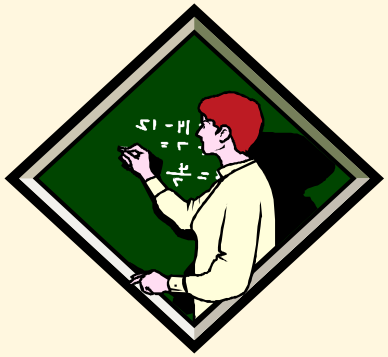
EPICS Base: IOC Toolkit

- Database, Sequencer, ChannelAccess.
- Plus: portability libraries
  - Linux, OS X, Win32, vxWorks, RTEMS, others.
- Basis for most of the "extensions"
  - Technically, sequencer is bundled as 'extension'.

- Probably the main difference between EPICS and other systems, where the support for front-end computers is often only helper libraries.

Record types

- Classified into four general types
- Input: e.g.
  - Analog In (AI)
  - Binary In (BI)
  - String In (SI)
- Algorithm/control: e.g.
  - Calculation (CALC)
  - Subroutine (SUB)
- Output: e.g.
  - Analog Out (AO)
  - Binary Out (BO)
- Custom: e.g.
  - Beam Position Monitor
  - Multi Channel Analyzer



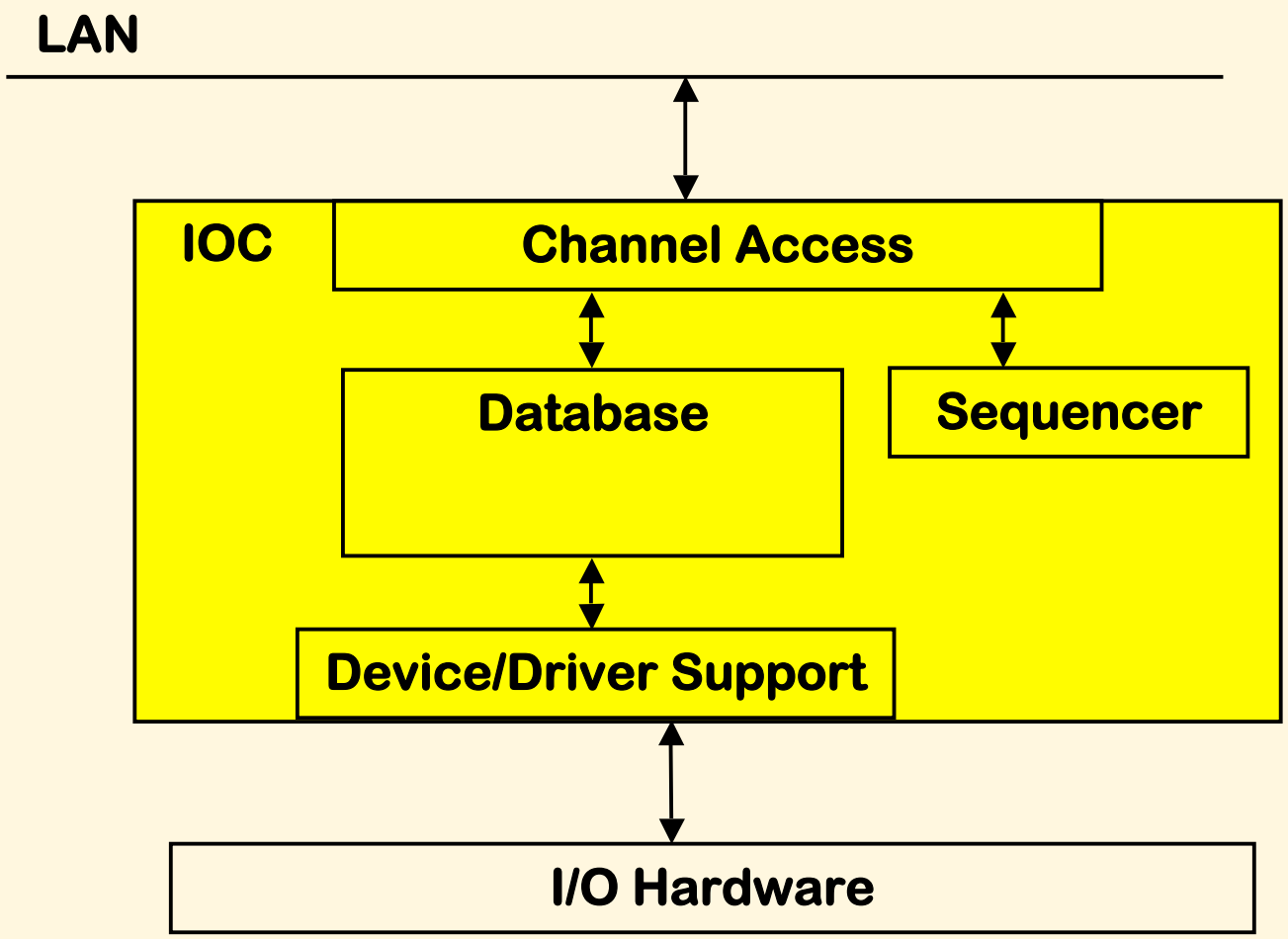
Some record types

- Analog in
- Analog out
- Binary in
- Binary out
- Calculation
- Calculation out
- Compression
- Data fanout
- Event
- Fanout
- Histogram
- Motor
- Multi bit binary input
- Multi bit binary output
- PID control
- Pulse counter
- Pulse delay
- Scan
- Select
- Sequence
- String in
- String out
- Subarray
- Subroutine
- Waveform

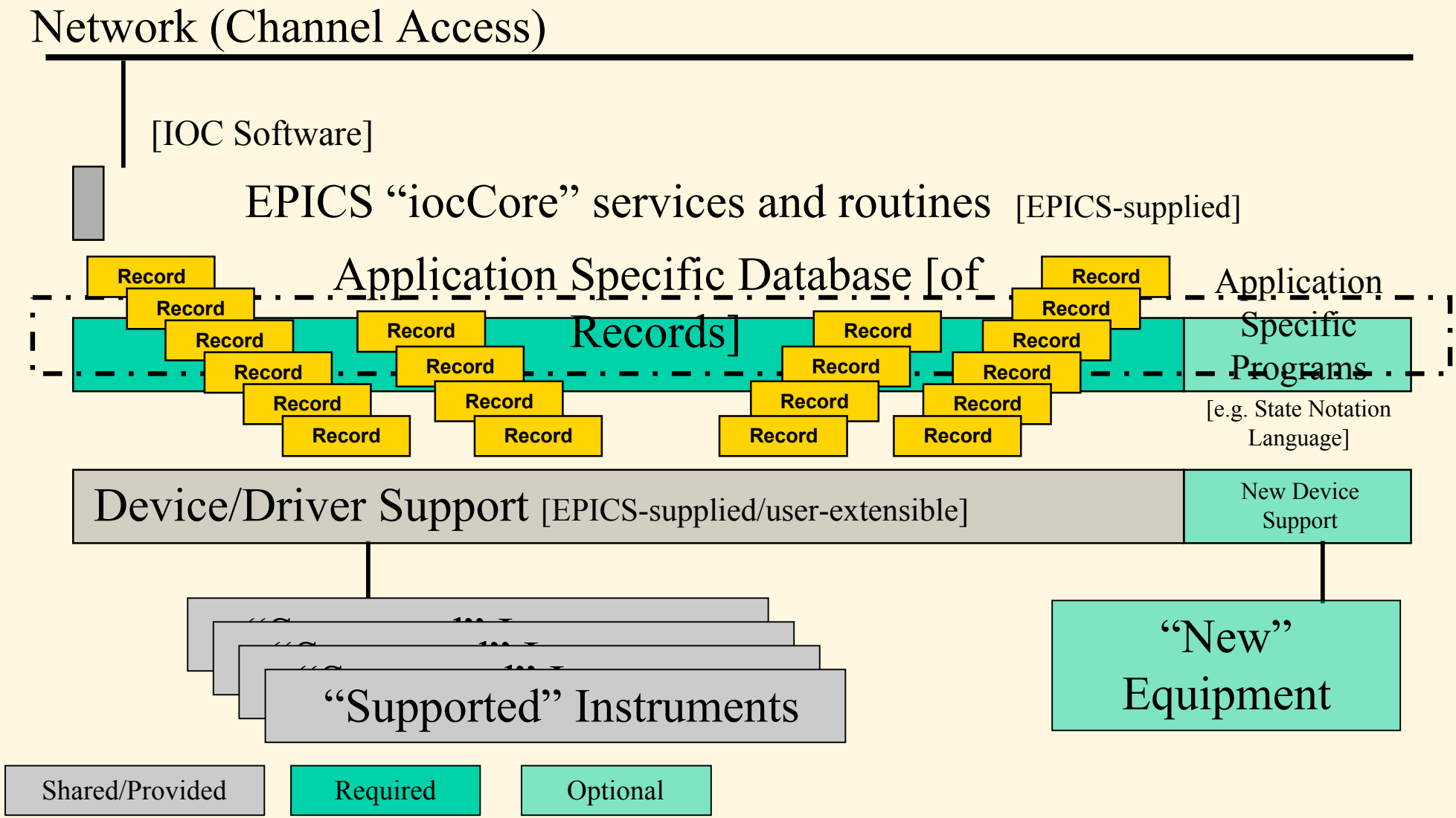
Record Processing

- Record processing can be periodic or event driven
- Periodic: Standard scan rates are…
  - 10, 5, 2, 1, 0.5, 0.2 and 0.1 seconds
  - Custom scan rates can be configured up to speeds allowed by operating system and hardware
- Event driven: Events include
  - Hardware interrupts
  - Request from another record via links
  - EPICS Events
  - Channel Access Puts

IOC Summary


LAN
IOC
Channel Access
Database
Sequencer
Device/Driver Support
I/O Hardware

IOC Software in One Slide



Device/Driver Support

- Software layer that interfaces records to hardware
  - EPICS web site lists many existing support modules.
  - Well defined Interfaces allow adaptation of new or custom hardware.
  - A recordType can interface to many different types of devices.
- AsynDriver is a facility that supports many different devices
  - Robust support for GPIB and Serial devices.
  - Also supports synchronous devices.

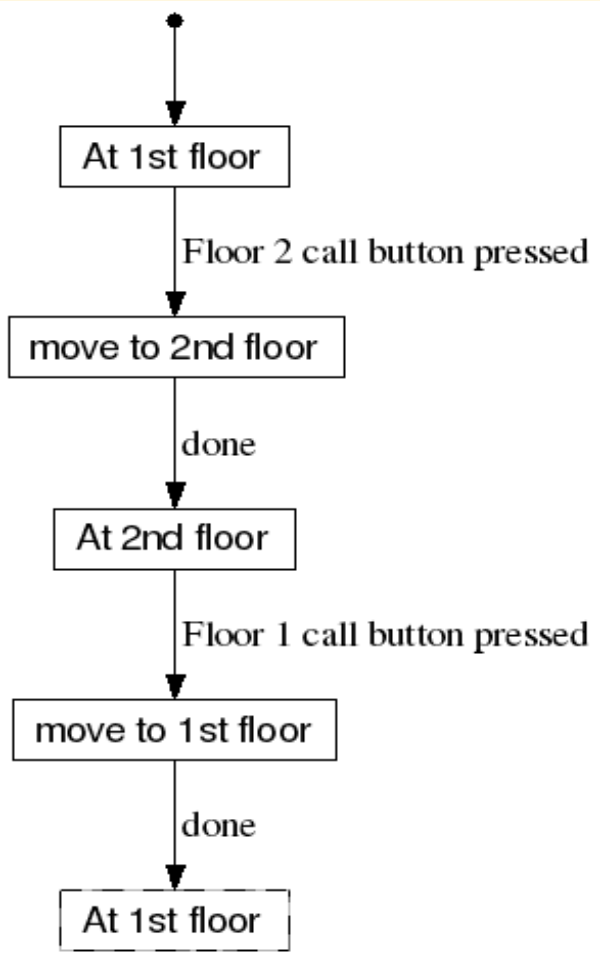
The Sequencer

- Runs programs written in State Notation Language (SNL)
- SNL is a ‘C’ like language to facilitate programming of sequential operations
- Fast execution - compiled code
- Programming interface to extend EPICS in the real-time environment
- Common uses
  - Provide automated start-up sequences like vacuum or RF where subsystems need coordination
  - Provide fault recovery or transition to a safe state
  - Provide automatic calibration of equipment

EPICS Sequencer

- Adds state-machine behavior to the IOC

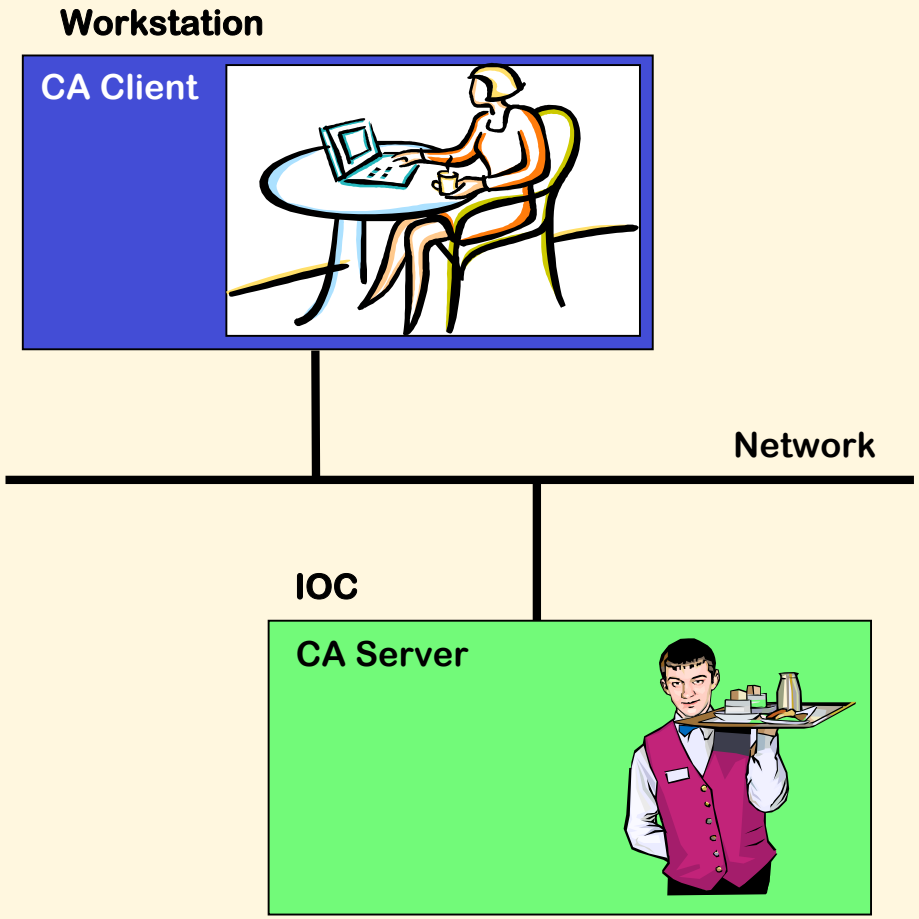
```
program Elevator_Simulation

ss Elevator
{
    state floor1
    {
        when (floor2_call)
        {
        } state goto2
    }

    state goto2
    {
     …
```



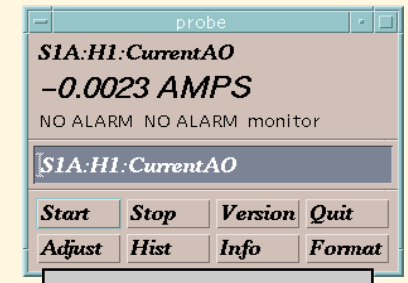
Channel Access

- Allows other programs (CA Clients) to see and change values of Process Variables in an IOC (CA Server)
- CA Clients may
  - Put (write)
  - Get (read)
  - Monitor

  data of Process Variables
- IOCs are both CA clients and CA servers. They can interact with data in other IOCs
- A CA Client can connect to many servers
- A CA Server may serve many clients
- A very efficient and reliable protocol



Channel Access Commands



Channel Access Client

CA Client

CA Server

Channel Access Server

Process Variables:

S1A:H1:CurrentAO

S1:P1:x

S1:P1:y

S1:G1:vacuum

"connection request" or "search request"

"get" or "caget"

"put" or "caput"

"set a monitor"

Who has a PV named " S1A:H1:CurrentAO"?

What is its value?

Change its value to 30.5

Notify me when the value changes

I do.

25.5 AMPS

OK, it is now 30.5

"put complete"

It is now 20.5 AMPS

It is now 10.5 AMPS

It is now 0.0023 AMPS

or

30.5 is too high. It is now set to the maximum value of 27.5.

or

You are not authorized to change this value

"post an event" or "post a monitor"

Some CA Clients

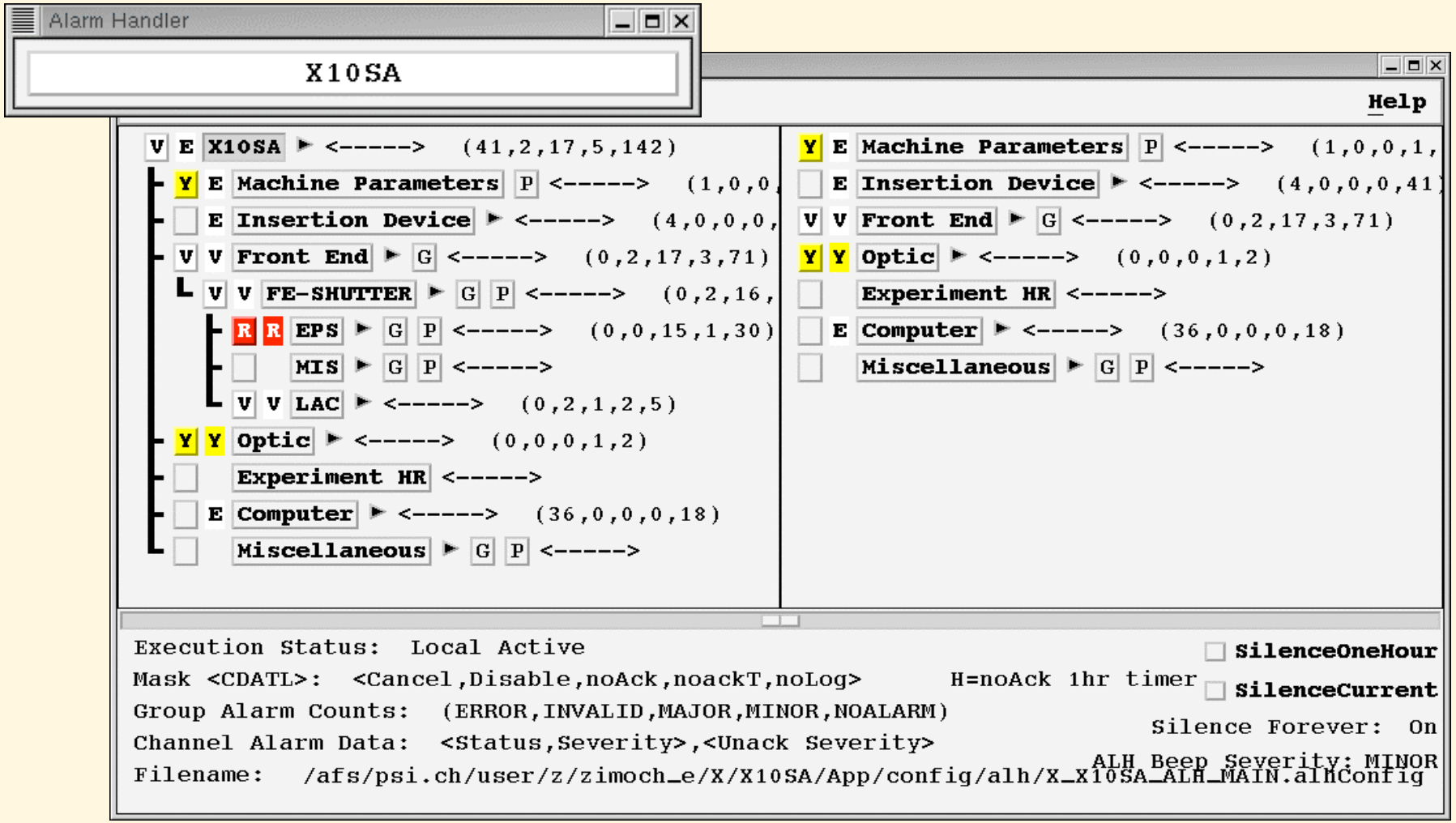
(from the EPICS Website - incomplete)

- ALH: Alarm Handler
- BURT: Backup and Restore Tool
- CASR: Host-based Save/Restore
- CAU: Channel Access Utility
- Channel Archiver (SNS)
- Channel Watcher (SLAC)
- EDM: Extensible Display Manager (ORNL)
- Knobs: Knob Manager und KnobConfig, eine Schnittstelle zu SunDials
- MEDM: Motif Editor und Display Manager
- StripTool: Strip-chart Plotting Tool
- and many more

Alarms and Colours

NO_ALARM	Green	Everything ok
MINOR	Yellow	Warning
MAJOR	Red	Error
INVALID	White	Device not reachable
Not connected	White block or Pink	Record not known

Example: ALH

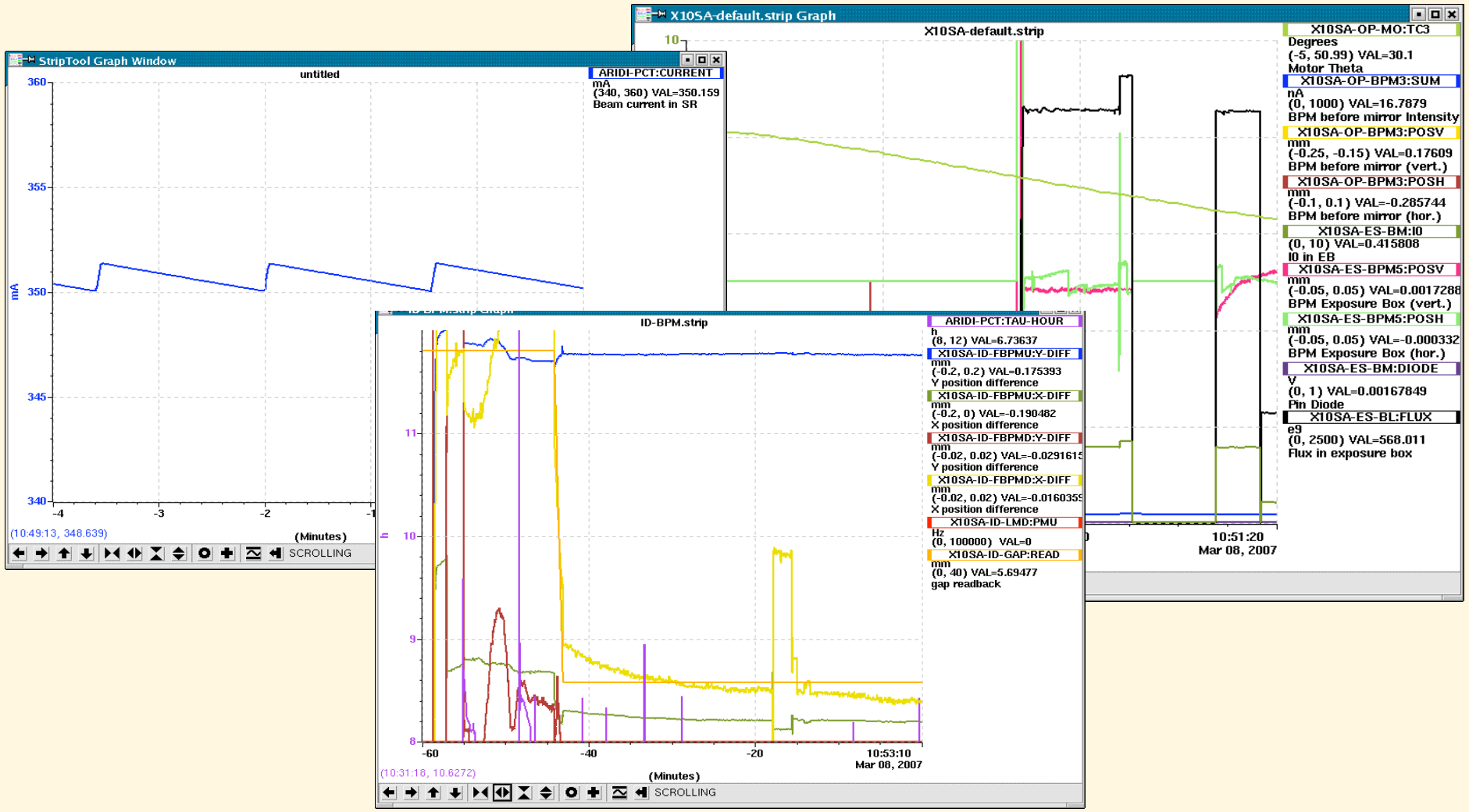

Alarm Handler
X10SA
Help
V E X10SA ► <-----> (41,2,17,5,142)
Y E Machine Parameters P <-----> (1,0,0,
E Insertion Device ► <-----> (4,0,0,0,
V V Front End ► G <-----> (0,2,17,3,71)
V V FE-SHUTTER ► G P <-----> (0,2,16,
R R EPS ► G P <-----> (0,0,15,1,30)
MIS ► G P <----->
V V LAC ► <-----> (0,2,1,2,5)
Y Y Optic ► <-----> (0,0,0,1,2)
Experiment HR <----->
E Computer ► <-----> (36,0,0,0,18)
Miscellaneous ► G P <----->
Y E Machine Parameters P <-----> (1,0,0,1,
E Insertion Device ► <-----> (4,0,0,0,41)
V V Front End ► G <-----> (0,2,17,3,71)
Y Y Optic ► <-----> (0,0,0,1,2)
Experiment HR <----->
E Computer ► <-----> (36,0,0,0,18)
Miscellaneous ► G P <----->
Execution Status: Local Active
Mask <CDATL>: <Cancel,Disable,noAck,noackT,noLog>
H=noAck 1hr timer
SilenceOneHour
SilenceCurrent
Group Alarm Counts: (ERROR,INVALID,MAJOR,MINOR,NOALARM)
Silence Forever: On
Channel Alarm Data: <Status,Severity>,<Unack Severity>
ALH Beep Severity: MINOR
Filename: /afs/psi.ch/user/z/zimoch_e/X/X10SA/App/config/alh/X_X10SA_ALH_MAIN.alhConfig

Some CA Clients

(from the EPICS Website - incomplete)

- ALH: Alarm Handler
- BURT: Backup and Restore Tool
- CASR: Host-based Save/Restore
- CAU: Channel Access Utility
- Channel Archiver (SNS)
- Channel Watcher (SLAC)
- EDM: Extensible Display Manager (ORNL)
- JoiMint: Java Operator Interface and Management INtegration Toolkit (DESY)
- Knobs: Knob Manager und KnobConfig, eine Schnittstelle zu SunDials
- MEDM: Motif Editor und Display Manager
- StripTool: Strip-chart Plotting Tool
- and many more ...

Examples: StripTool

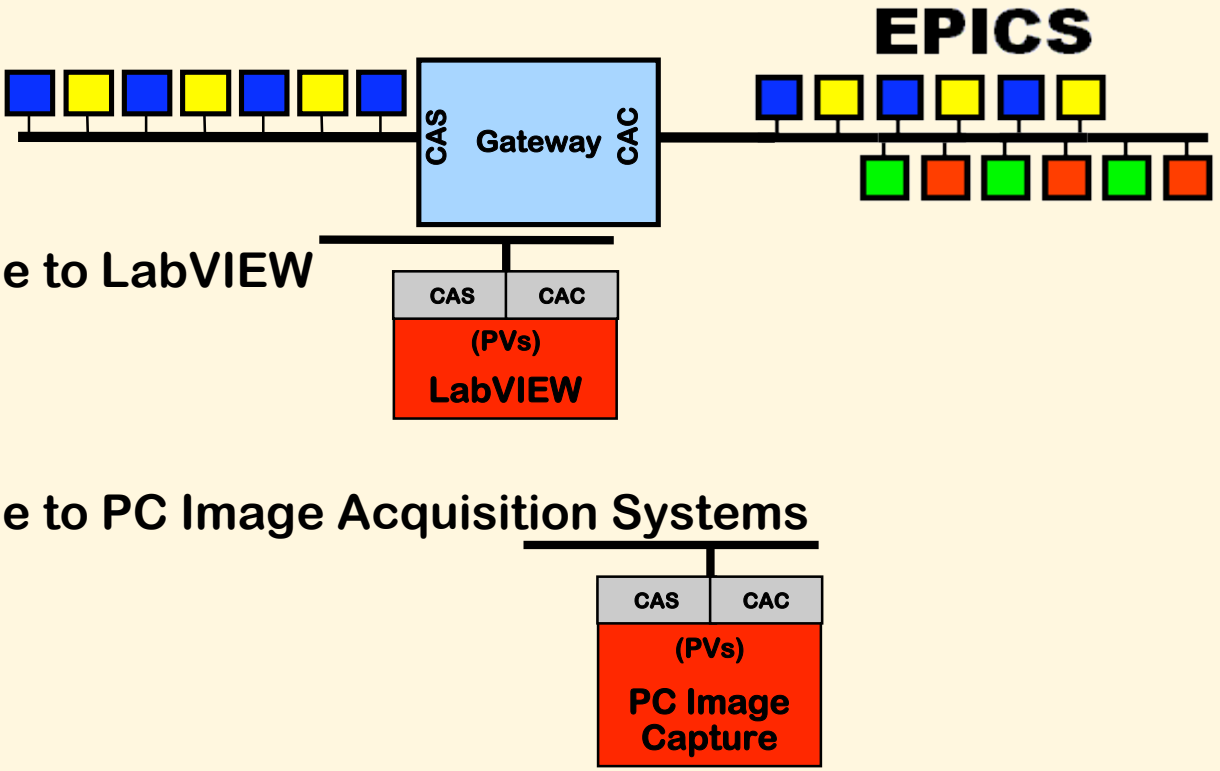


Control System Studio

- CSS – Control System Studio
  - Matthias Clausen (DESY-Germany) started CSS
  - In Google ask for Control System Studio
  - Eclipse Based Plug-in Environment
  - Basic plug-ins already available.
  - Separate User and Developer Environments.
- But CSS
  - Is more than just an Application.
  - Is more than a container for a selection of (Eclipse) plugins.
  - Is more than the adaptor for plugins which (more or less) comply with the CSS 'standards'
  - Is an idea.
  - Read more about it!!!!

Popular CA Server Applications

- IOC Core
- PV Gateway
- CA Server Interface to LabVIEW
- CA Server Interface to PC Image Acquisition Systems



How fast is EPICS?

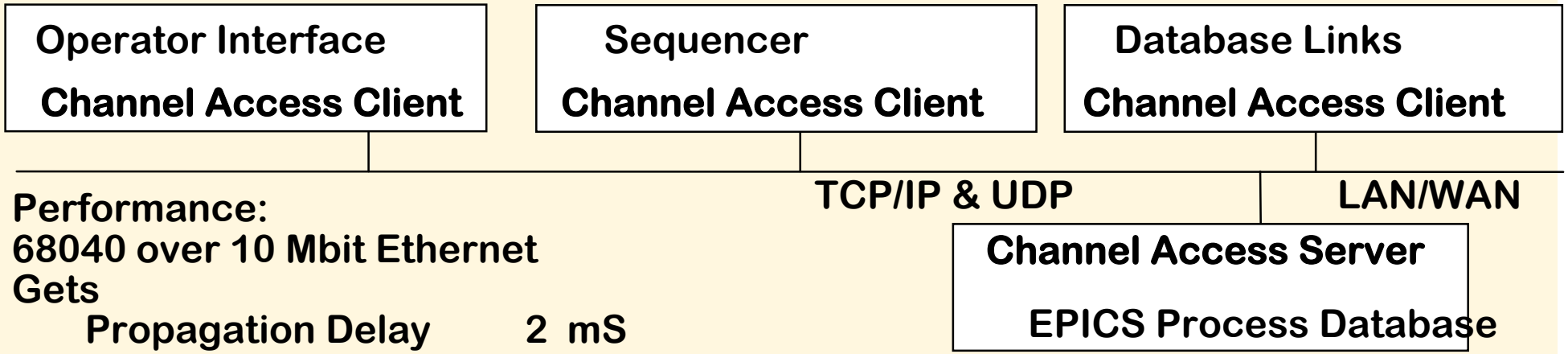
- Can be fast or slow, it depends how you use it!
- Use the correct tool for the job; Database, sequencer, custom code (ioc) or custom code (client)
- Ultimately speed depends upon hardware
- Some benchmarks*:

Machine	OS	CPU	Speed	Rec/sec	%CPU
MVME167	vxWorks	68040	33MHz	3,000	25
MVME 2306	vxWorks	PPC604	300MHz	20,000	20
MVME5100	vxWorks	PPC750	450MHz	100,000	25
PC	Linux	PII	233MHz	10,000	27
PC	Linux	P4	2.4GHz	100,000	18

* Extrapolated from benchmark figures courtesy of Steve Hunt (PSI) and L.Hoff, (BNL)

- Modern GHz CPU processes a record in 1-to-several microseconds
- Database design and periodic scanning effect *apparent* system speed

IOC Core: Channel Access Services

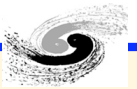


Performance:
68040 over 10 Mbit Ethernet
Gets
- Propagation Delay 2 mS
- Throughput 7.8K /sec

Puts
- Propagation Delay 1 mS
- Throughput 17K /sec

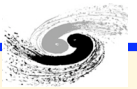
Monitors
- Propagation Delay Dependent
- Throughput 15K / sec
- (Typically 10% channels have monitors)
- (memory use in IOC - 2 Meg / 60 connections)
- (30% network load for 15K monitors / second)

Increase bandwidth with Routers, Bridges, Higher speed networks and EPICS gateway



Ten really neat things about EPICS

- It's free
- It's Open Source
- There are lots of users
- All a client needs to know to access data is a PV name
- You can pick the best tools out there …
- … or build your own
- The boring stuff is already done
- There is a lot of expertise available close by
- A good contribution becomes internationally known
- By following a few simple rules, you get a lot for free



The Learning Curve for EPICS is difficult

- Installing EPICS
- Setting up the application environment to automatically build databases
- Setting up the IOC to boot from the workstation
- Installing the new drivers
- Knowledge of how to debug the application - is needed by everyone
- Learning to use the process database
- Choosing and learning which client tools to use

- This learning curve can be eased by receiving training from other laboratories, having one of your employees work and train at an EPICS site, or reading the documents and using the software support document to determine the collaboration member supporting your platform.