



Argonne  
NATIONAL  
LABORATORY

... for a brighter future

## *Introduction to EPICS*

(Ned Arnold, John Maclean)

Modified by Michael Davidsaver



U.S. Department  
of Energy

UChicago ►  
Argonne<sub>LLC</sub>



A U.S. Department of Energy laboratory  
managed by UChicago Argonne, LLC

## *What is EPICS?*

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

# What is EPICS?

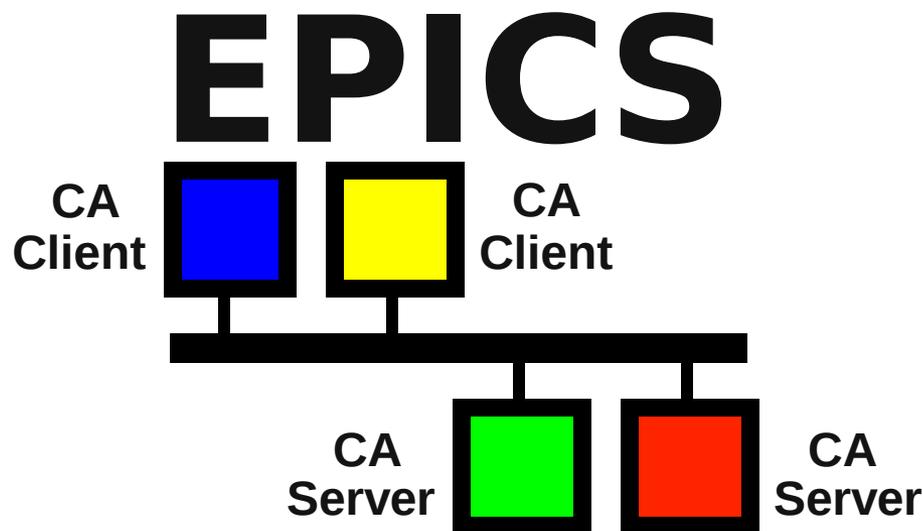
- A Collaboration
  - Began in 1989 between LANL/GTA & ANL/APS
    - *(Bob Dalesio & Marty Kraimer)*
  - Over 150 license agreements were signed *before* EPICS became “open source”
  - Recent EPICS collaboration meetings
    - *100+ Attendees*
    - *30+ Institutions*
    - *75+ Presentations*
  - List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

# What is EPICS?

- Major Collaborators
  - ANL (APS Accelerator, APS Beamlines, IPNS)
  - LANL
  - ORNL (SNS)
  - SLAC (SSRL, LCLS)
  - JLAB (CEBAF)
  - DESY
  - BESSY
  - PSI (SLS)
  - KEK
- Recent Collaborators
  - *DIAMOND Light Source* (Rutherford Appleton Laboratory, Oxfordshire)
  - *The Australian Synchrotron (AusSy)* (Melbourne)
  - *BNL (NSLSII)*

## What is EPICS?

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

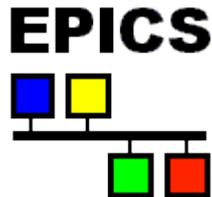


For EPICS, *client* and *server* speak of their Channel Access role

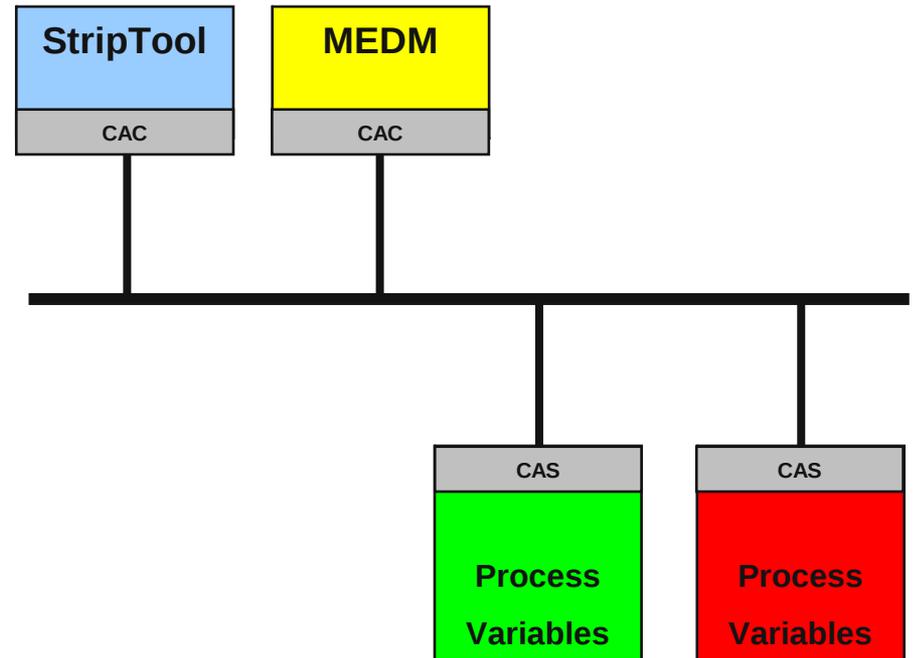
i.e. Channel Access Client & Channel Access Server

# 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?

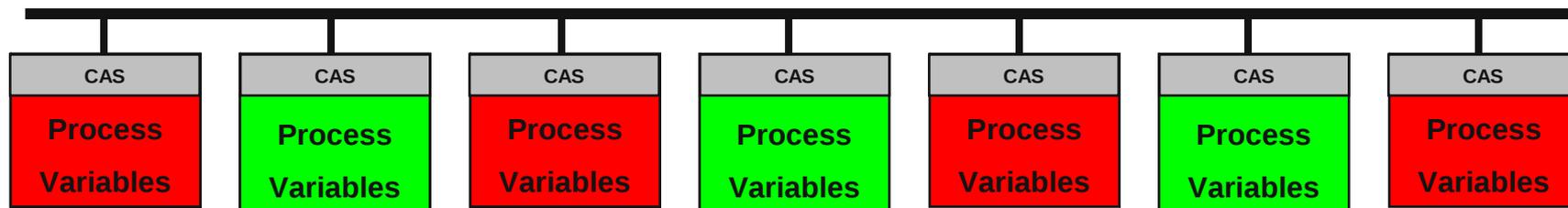
- Process Variable
  - A ***Process Variable*** (PV) is a named piece of data associated with the machine (e.g. status, readback, setpoint, parameter)
  - Examples of PV names and **values**:
    - *S1:VAC:reading* 3.2e-08 torr
    - *LINAC:BPM4:xPosition* -0.323 mm
    - *BOOSTER:gateValvePosition* '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)*
    - *Timestamp*
    - *Normal Operating Range*
    - *Control Limits*
    - *Engineering Unit Designation (e.g. degrees, mm, MW)*

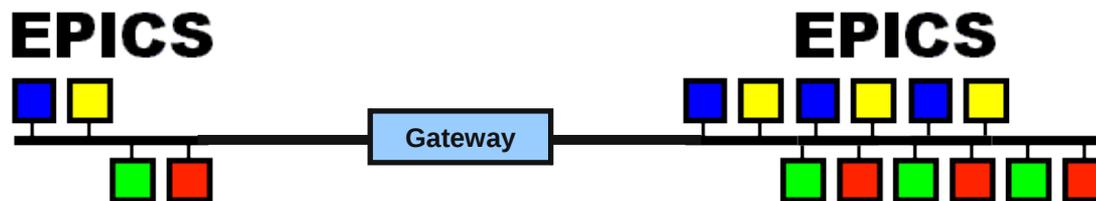
# What is EPICS?

- 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 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

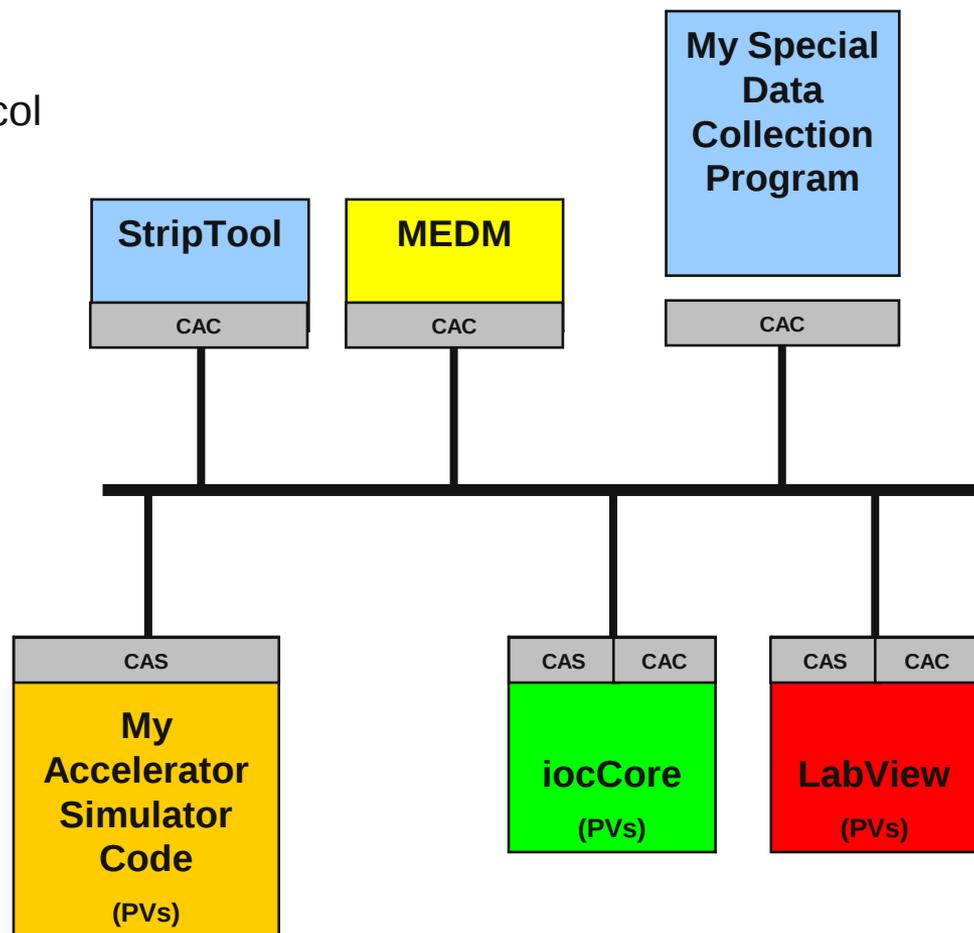


## What is EPICS?

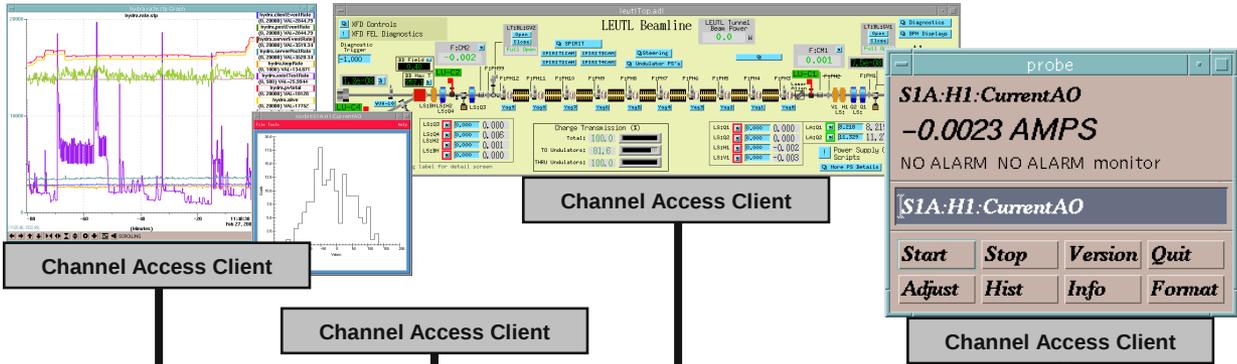
- Any tool/program/application that abides by the Channel Access protocol could be described as “EPICS Compliant”.

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

Python, Matlab, Java, etc.



# How does it do it?



Channel Access Client

Channel Access Client

Channel Access Client

Channel Access Client

**Channel Access Server**

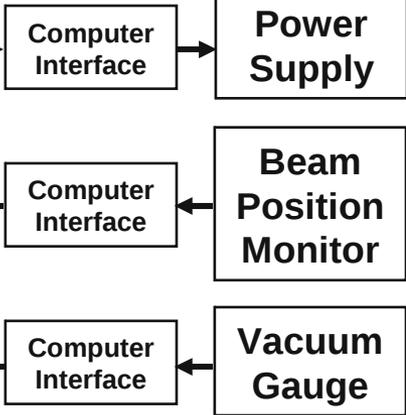
**Process Variables:**

S1A:H1:CurrentAO

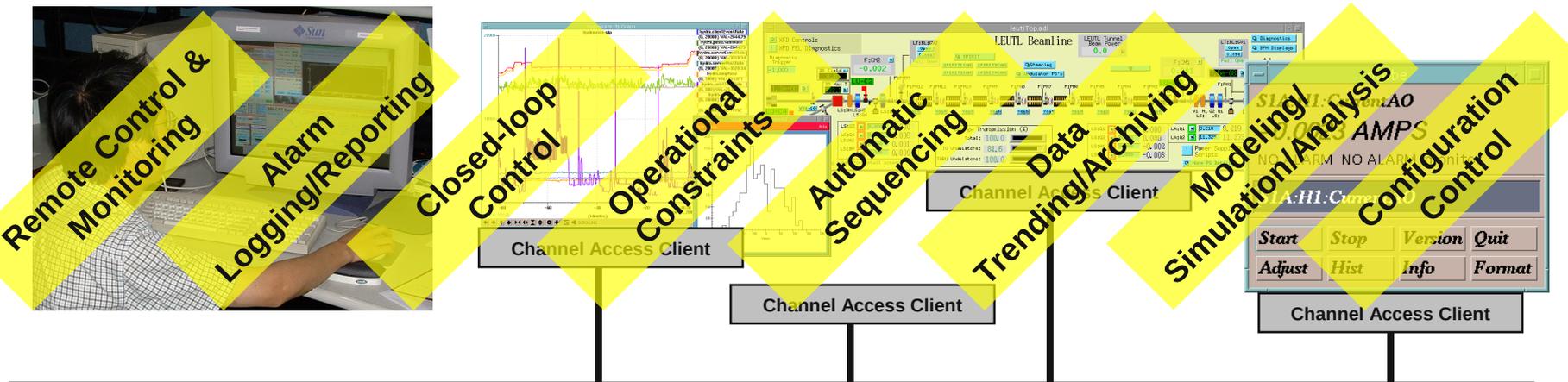
S1:P1:x

S1:P1:y

S1:G1:vacuum



# Where does it do it?



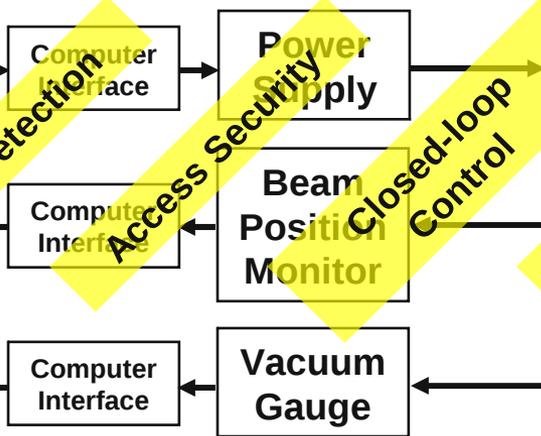
**Channel Access Server**

**Process Variables:**

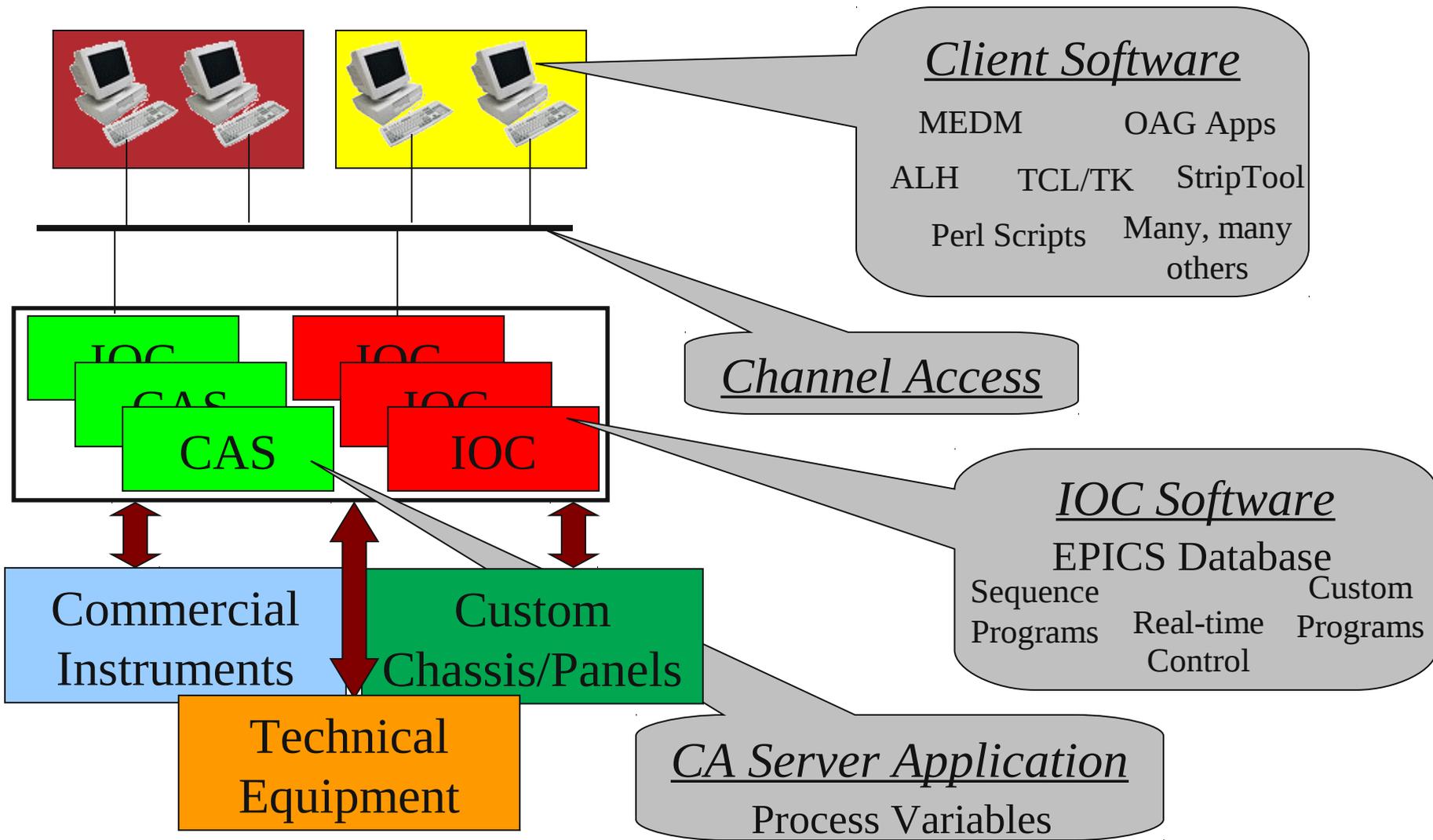
- S1A:H1:Current AO
- S1:P1
- S1:I1
- S1:G1:vacuum

**Data Conversion/Filtering**

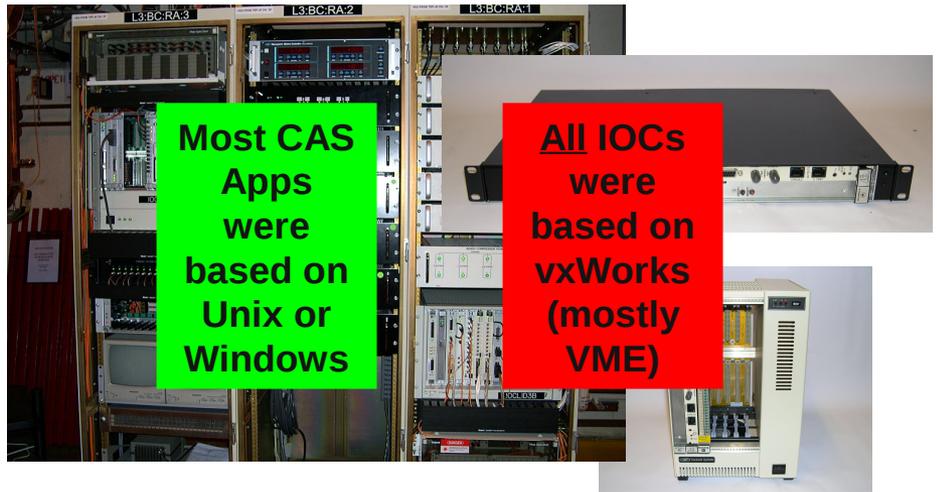
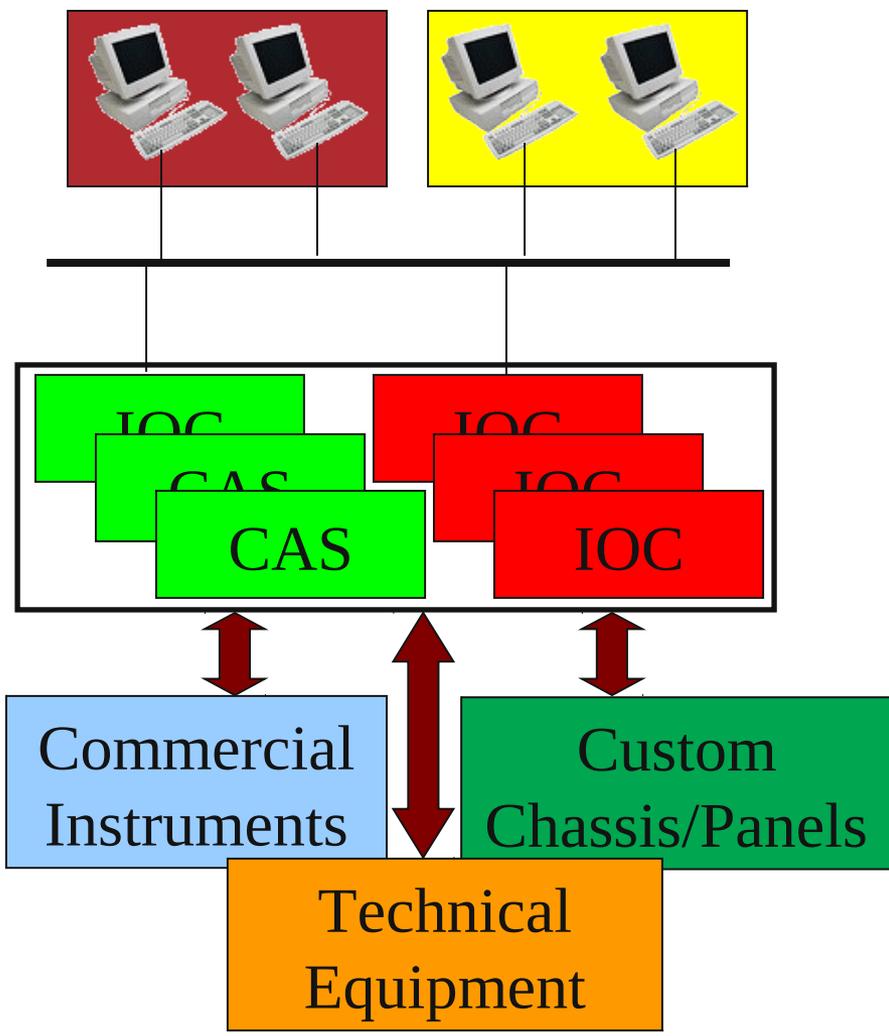
**Alarm Detection**



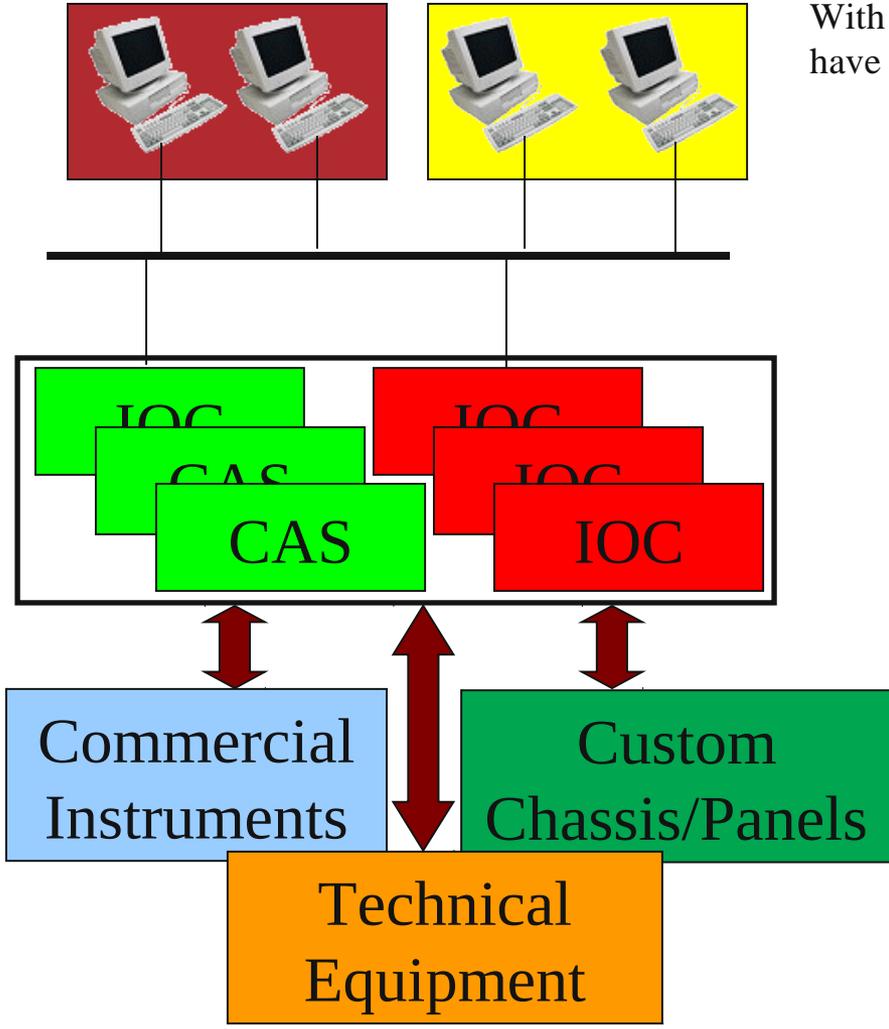
# Canonical Form of an EPICS Control System



# 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.



## 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)

- **Database design and periodic scanning effect *apparent* system speed**