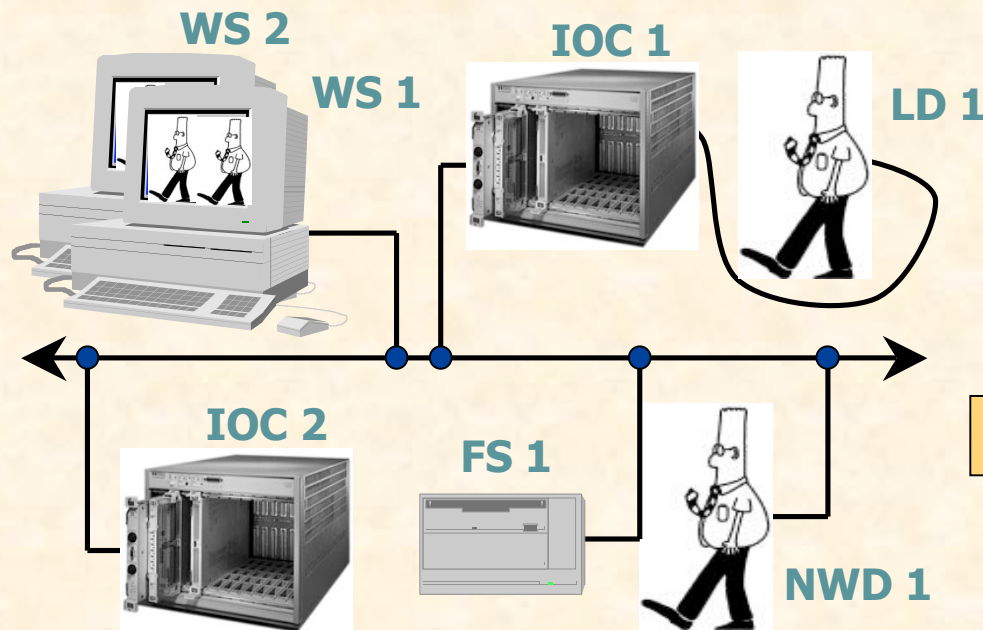


Make ***“This”*** work with EPICS!

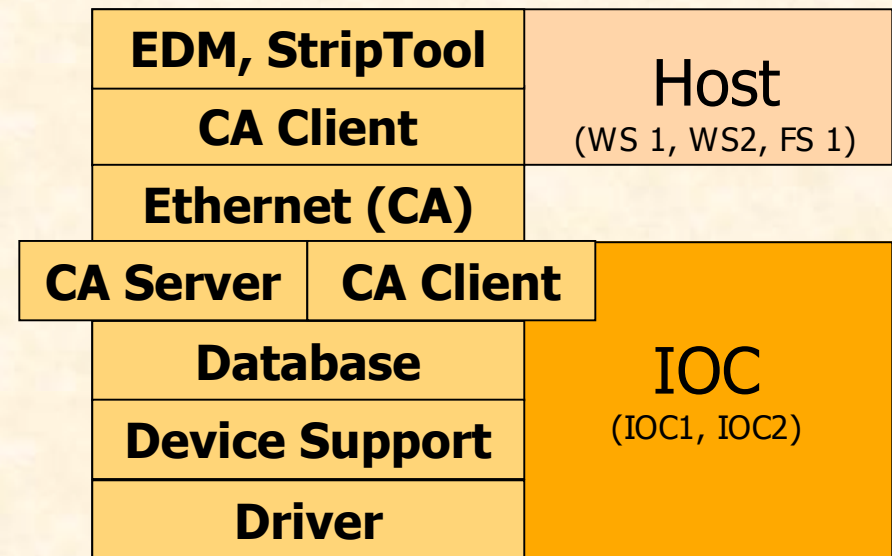
2006

kasemirk@ornl.gov

Where does new “This” fit in?



EPICS Block Diagram



Low-Level Driver

- Typically in C, C++
- Provides basic methods to find, initialize, read, write the device
- Usually it's not specific to EPICS or anything in particular except the OS
 - vxWorks, RTEMS, Linux, Win32, ...
- Vendor of device might already provide it.
- If not: Consider use of EPICS 'OSI' routines
 - Makes it specific to EPICS, but offers portability across OSs.

This=VME/VXI/ISA/PCI/cPCI board?

- **Put into VME crate or PC**
- **Load EPICS base software**
- **Connect low-level driver to iocCode:**
 - a) **Add subroutine record 'init' and 'process' routines.**
 - **Initially easy, but sub record doesn't add much.**
 - b) **Add SNL code, invoke low-level driver from within states.**
 - **Works quite well for one-off setups, but can result in a mess if used without restraint.**
 - b) **Add device support for existing record types**
 - **More initial work, but benefits from AI, BI, ... functionality, and results in "standard" setup that others might best understand.**
 - c) **Add new record types specific to the device**
 - **Don't know an example where this worked out OK.**

This=Something else

- **Since it's not physically in the front-end computer, and in the past only vxWorks-based IOCs could run iocCore, the answer was:**
Connect it to the EPICS network via CA
 - **Write custom CA server**
 - **Or use CA client lib to 'push' data into an IOC.**
- **Since EPICS R3.14, iocCore runs on most platforms**
 - **Can use subroutine rec, device support, SNL, ...**

Acknowledgements

- **Material has been copied from**
 - **Martin Pieck (LANL)**
- **Ideas**
 - **Bob Dalesio and many others**