Brookhaven™
National Laboratory

U.S. DEPARTMENT OF
ENERGY

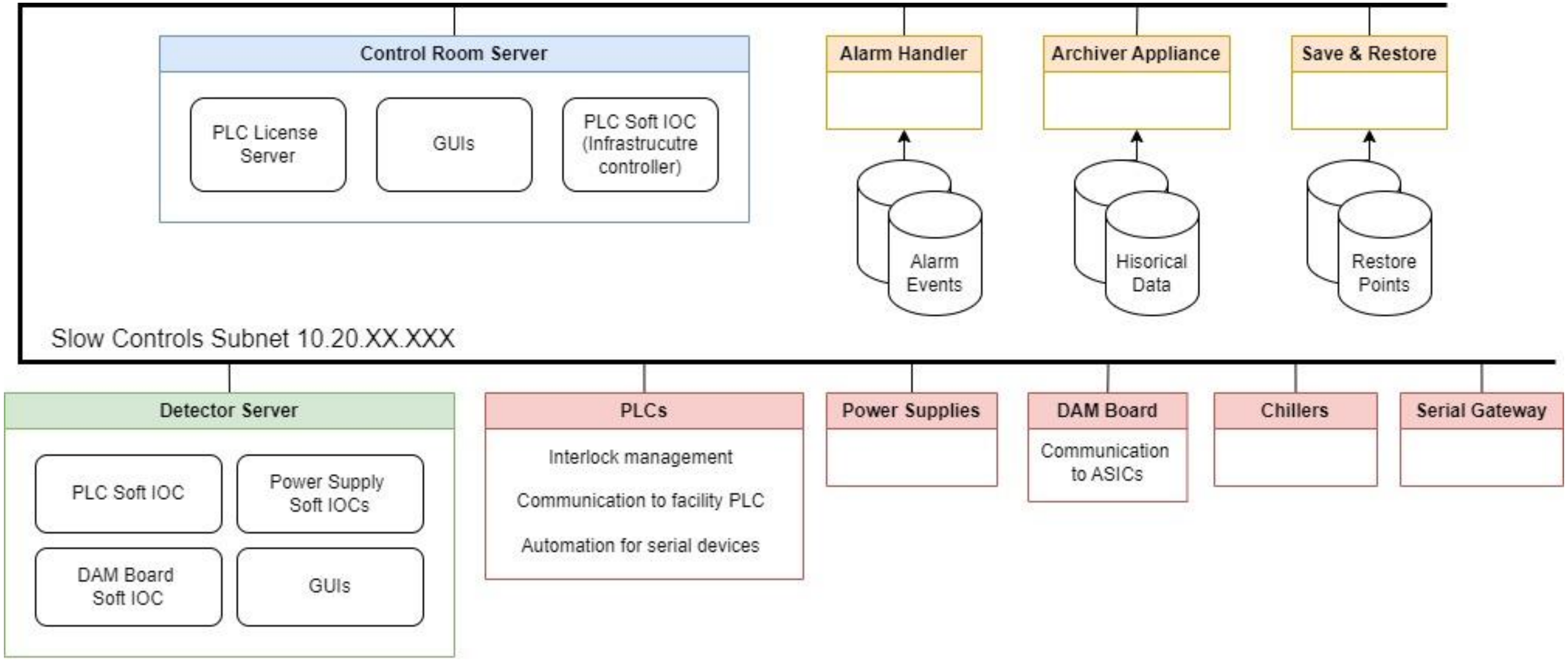
ePIC Slow Controls Prototyping

@BrookhavenLab

Agenda

1. Slow Controls Approach
2. Controller Selection
3. Possible Final Implementation
4. Prototyping

Slow Controls Approach



Controller Selection

Allen Bradley Controller Families

- ControlLogix
  - Highest program memory
  - Typically for large system
  - Supports remote IO
  - Overkill for a single detector
- CompactLogix
  - Moderate program memory
  - Typically used for small systems
  - Supports remote IO
- Micro800 (NOT MicroLogix!!!)
  - Small program memory
  - Typically used for single machines
  - Remote IO not supported
  - Inexpensive
  - Looks good on paper, but untested

EPICS support

No EPICS support, yet.
A way to save during prototyping? Maybe…

Controller Selection

Allen Bradley Remote IO Families

- Point IO
    - 64 IO modules per node
        - Analog IO
        - Digital IO (sourcing and sinking)
        - Relay modules
        - RS 232/485 communication
        - RTD and thermocouple modules
- Flex5000
    - Higher IO density per module, fewer modules per node
    - Digital IO
    - Analog IO
    - No specialized modules

Controller Selection

Allen Bradley Redundancy Solutions

- Device Level Ring Networks (DLRs)
  - Like a token ring
  - Device managed ring network with a minimum of 1 supervisor node
    - Supervisor will optimize traffic through the ring and alarm in the event of an interruption
  - Possible supervisors
    - Allen Bradley Stratix Switch (a bit pricey, but it's a backbone Cisco managed switch)
    - Your PLC (limits you to a copper network)
    - Allen Bradley ETAP (copper to fiber converter)
- Why should we use a DLR if there are only 2 nodes in the ring?
  - The PLC can be kept in the counting house, a much cleaner environment
  - If the supervisor node is the PLC or PLC adjacent, you will be alerted of a communication break
  - Remote IO behavior can be preprogrammed in case of a communications fault

Controller Selection

Controller Software

- PLC development
    - Studio 5000
        - Needed for ControlLogix and CompactLogix PLCs
    - Connected Components Workbench
        - Needed for Micro800 PLCs
- GUI development
    - FactoryTalk View
        - Allen Bradley product
        - Works fine, needs a license
    - Ignition
        - Generic GUI developer for PLCs
        - Free trial version
        - Modular license model
    - EPICS
        - FREE
        - Learning curve
        - What will be used at the end of the day
    - Python!
        - Pylogix can interface with ControlLogix, CompactLogix, and Micro800 PLCs (how convenient!)

Possible Final Implementation

- PLC: CompactLogix Controller
- Remote IO: Point IO Ethernet Module (1734-AENT) and supporting IO
- Redundancy measures: Allen Bradley 1734-ETAP or 1734-ETAP2F
- Miscellaneous hardware:
  - Switches, power supplies, sensors & actuators, etc.
- Supporting Software:
  - Studio 5000
  - EPICS

Prototyping with Final Implementation

Pros

- Portable

Cons

- $$$
- Software and firmware licensing issues
- Different coding styles

Prototyping on the cheap with AB

- Controller: Micro800 PLC (recommend a Micro850)
  - Comes with 10 digital inputs and 10 digital outputs built in
  - Expansion modules available for:
    - Analog IO
    - RTD/thermocouples
    - Serial communication
    - Motion controls
- No RIO or redundancy
- Misc. Hardware
- Supporting Software
  - Connected Components Workbench
  - GUI of choice

Resources for controller selection

- Integrated Architecture Builder
  - Free software from Allen Bradley
  - Use wizards to determine a PLC configuration and BOM

Questions, Comments, Concerns