WBS 2.7.4 Breakout Detector Controls (DCS)

A.Norman (U.Virginia)
L3 Manager

June 16, 2009

Detector Controls & Monitoring: Overview

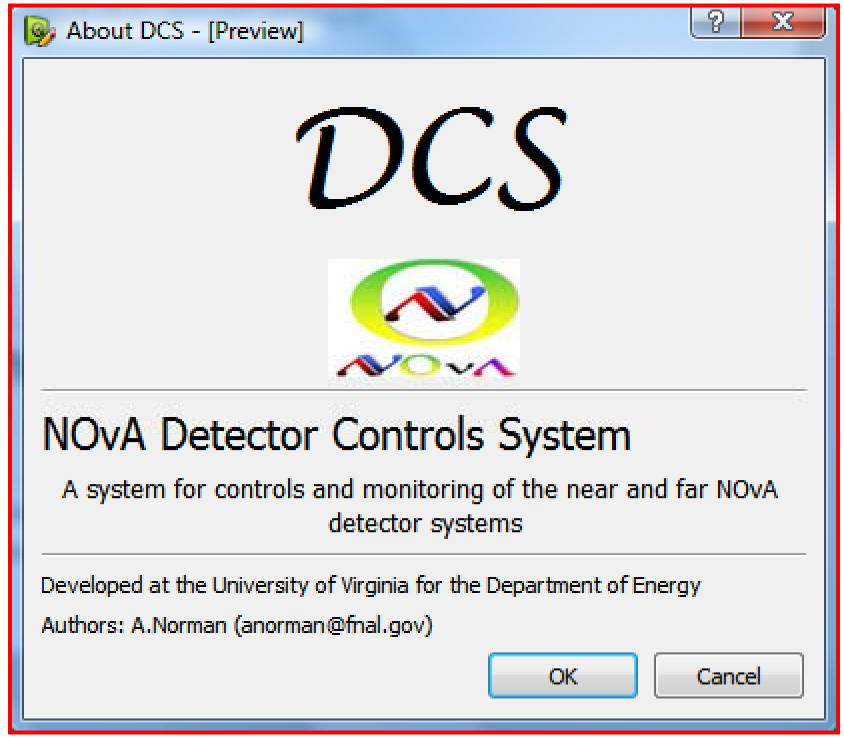
- What is the "D.C.S"?
  - It is the *unified* collection of detector controls, readout monitoring systems
  - It controls all non-DAQ specific configuration parameters for hardware systems
  - It controls some DAQ specific configuration parameters, but only for ***hardware*** systems
  - It monitors all critical operational parameters
  - It handles the generation of alarm and logging messages for the system using these parameters
  - It interfaces to the main databases
- This is the User's (and Run Control's) window into controlling NOνA

Detector Controls & Monitoring: Status

- Full system design established
  - Heavily leveraged off of EPICs libraries
  - Designs for specific hardware readout exist where hardware has been established
  - Preliminary [generic] designs for hardware readout exist where hardware has not been finalized
  - Prototype [standalone] readouts have been developed or leveraged for key systems still being evaluated
  - Implementations of some subsystems have been started
    - Completion requires final hardware choices
    - Completion requires final DAQ interaction specifications
- DCS design is approximately 90% complete

Overview

- Scope
- Hardware
- Computing
- Networking
- Software Design
- Integration



Detector Controls & Monitoring

SCOPE

Scope: Controls

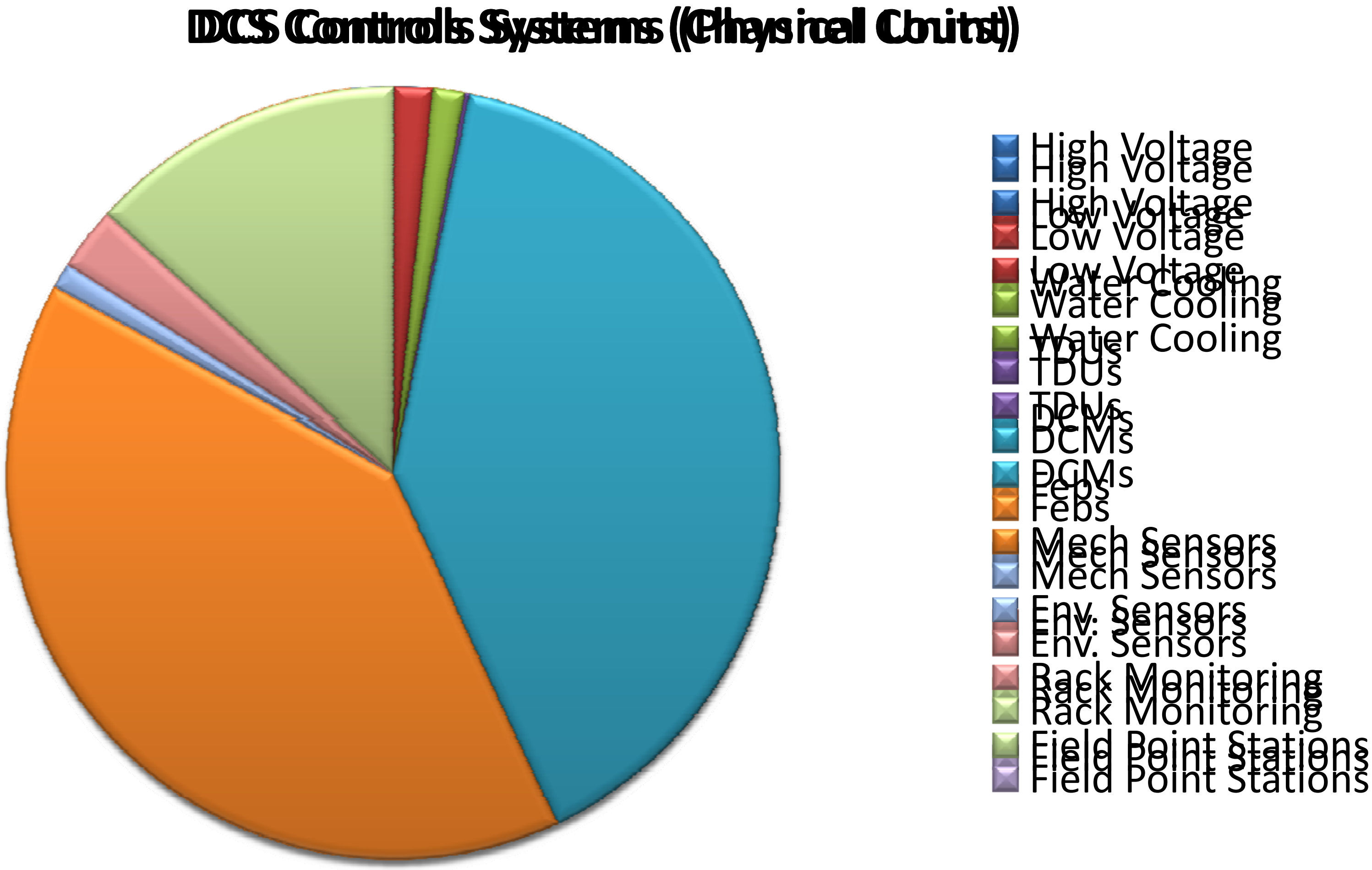
- The DCS is designed to handle the programmable controls, monitoring, alarm handling, and state logging for the major detector subsystems
  - Power Control and Distribution
    - High Voltage Bias
    - Low Voltage power
  - Water pumping and cooling
  - Data Concentrator Modules (DCM)
  - Front End Boards (FEB)
  - PVC Stress/strain sensors
  - Timing/Sync modules
  - Environmental parameters and monitoring
  - Rack monitoring
- The DCS needs to provide the local state logging as well as interfacing with the primary database server for long term logging and system history
- The DCS needs to provide a feedback loop into the DAQ system to handle system faults and run stop conditions

Scope: Far Det. Controls (15kTon Design)

Subsystem	Type	Units	Channels	Total Channels	Interface
High Voltage	Caen SY1527 or Weiner Mpod	2	96	180 used	Ethernet
Low Voltage	Wiener PL506 or Caen SY8800	60	6	360	Ethernet
Water Cooling	Neslabs M150	15	10	150	Ethernet/RS485
Data Acq.	Custom DCM	180	62(64)	11,160	Ethernet
Det. Envir.	N.I. Fieldpoint	15	128	1,920	Ethernet
Front End	Custom FEB	11,160	32	357,120	8B/10B DCM
Timing/Sync	Custom TDB	12	4	48	Ethernet
Misc. & Rack Mon.	Misc. Sensors	200	1-10	2000	Misc.
			Total	15,818 372,938 w/FEBs	

- The detector controls system is required to have access to over 11,600 physical devices roughly 370,000 independent programmable channels
- Controls interface must present a uniform method of accessing all the devices regardless of controls interface
- The system must be modular, scalable and support partitioning into production, installation, and calibration/commissioning variants to accommodate the “physics during build” model.

Far Det. Controls (14kTon)



Scope: Far Detector Monitoring

System	Parameters/Chan	Channels	Total Monitoring
High Voltage	6	180	1,080
Low Voltage	4	360	1,440
Water Cooling	4	150	600
DCM	24	180	4,320
Det. Env	128	15	1920
Front End	32 max	11,160	357,120
Timing/Sync	10	12	120
Mech Sensors	1	4500	4500
Misc. & Rack Mon.	1-2	150	300
		TOTAL	14,280 371,400 w/FEBs

- The detector Monitoring system is required to monitor and alarm on over 370,000 system critical run time parameters.
- The monitoring frequency must be variable from ~1Hz for pushed data streams to once per minute (or 5minutes) for slower polled readouts.
- Client initiated monitoring requests can cause burst spikes in the DCS bandwidth with transfers on the order of 400-600mb per full monitoring dump with protocol and network overhead

Detector Controls

HARDWARE

DCS Readout Hardware

DCS Readout interfaces fall into two general categories

Ethernet Enabled

- CAEN SY1527High Voltage
- Wiener PL508 Low Voltage
- Data Concentrator Modules
- Compact Fieldpoint Controllers
- Remote power protection circuits

These we readout directly via network switch connection

Non-Ethernet Enabled

- Temperature sensors (RTDs)
- Pressure and flow sensors
- Stress/Strain gages
- Local relays
- Rack monitoring
- Misc sensor packs
- RS485 protocol devices
- Front End Boards (FEBs)

FEBs are readout via a DCM pass through method

These we readout first by a Fieldpoint station, then into the network

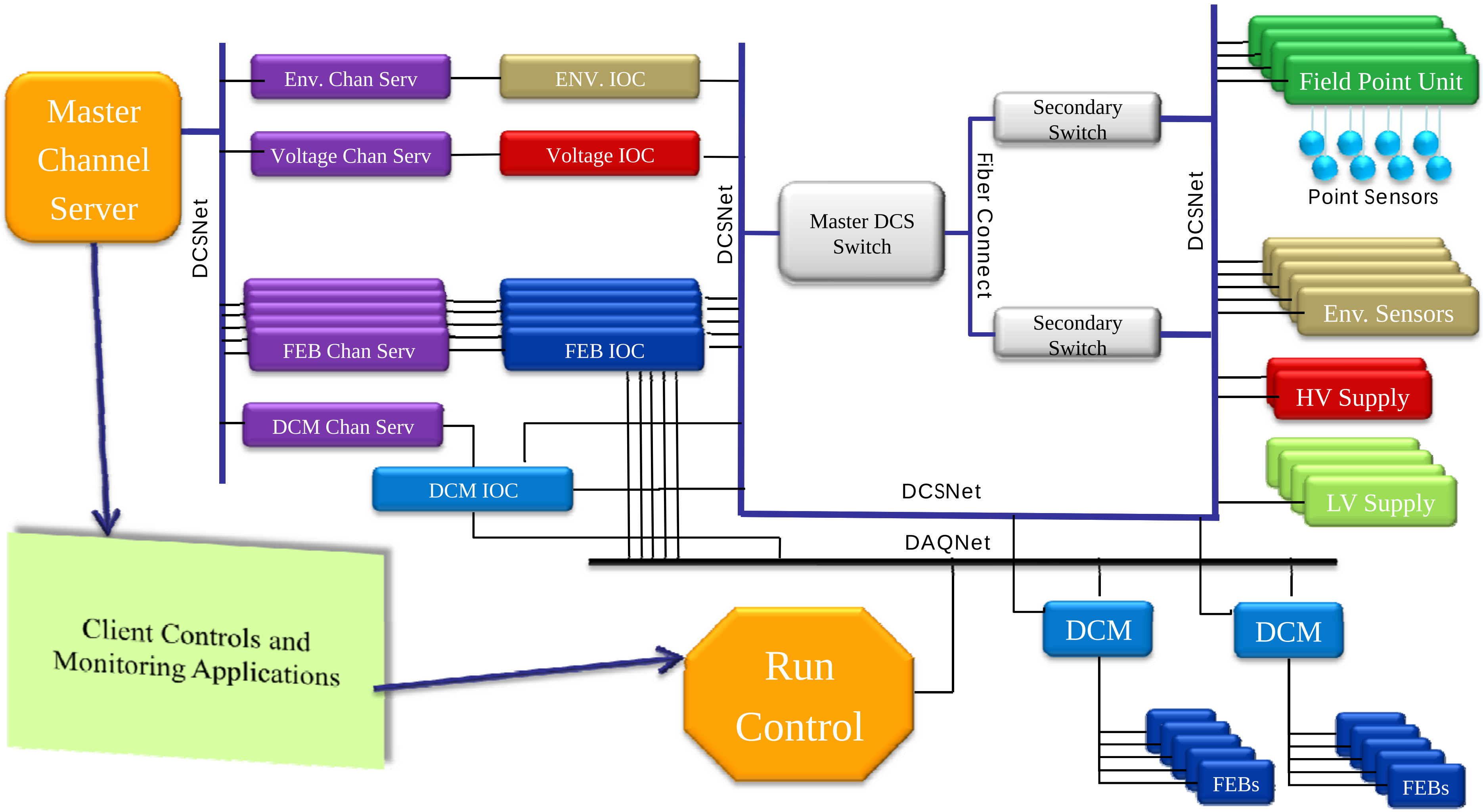
DCS Readout Hardware

- Ethernet enabled devices are connected directly to a second level switch and read through the DCSNet
- Sensor packs are readout through a National Instruments Compact Field Point station which is outfitted with an Ethernet control interface
- Front End Boards are readout through the DCMs in a pass through fashion that routes the status packets through embedded system handler
  - DAQ/Data traffice is routed to Eth0 to DAQNet
  - DCS traffic is routed out on Eth1 to DCSNet

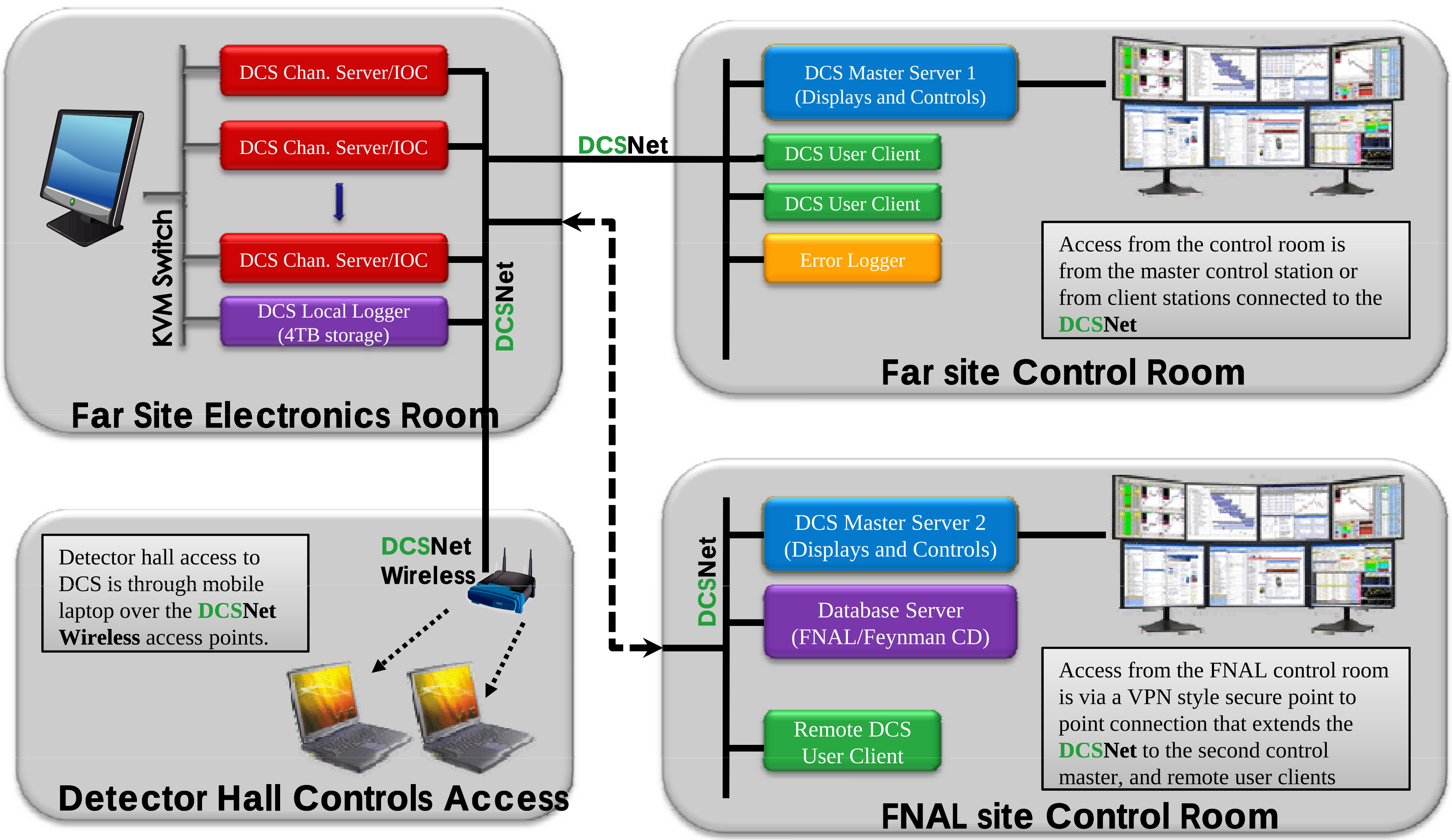
Fully Partitioned Network Traffic

- The general monitoring stations are arrayed in the detector hall on the upper catwalk spaced by di-block

DCSNet Network Topology



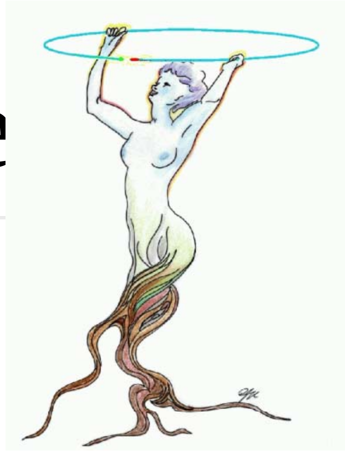
DCS Network Overview



Detector Controls

SOFTWARE

DCS Client Software



- Based off of QT4 and ROOT 5 with interfaces to EPICS 3.14
- Currently working under QT4.2.x and QT-ROOT 5.x
- Runs under Sci Linux 4.x and all Microsoft Windows XP/Vista distributions

- Provides a platform independent development API for creating the client interfaces.
- Client applications are portable and can be deployed on multiple platforms
- Everything is standard C++ compatible with standard GCC (linux) and Visual C/mingw (MS Windows)
- Results in high performance clients

- Visualization is accomplished by direct use of the ROOT histogram and plotting libraries.
- Component widgets are standardized and manageable.
- Full integration with ROOT 5.x complete under QT4

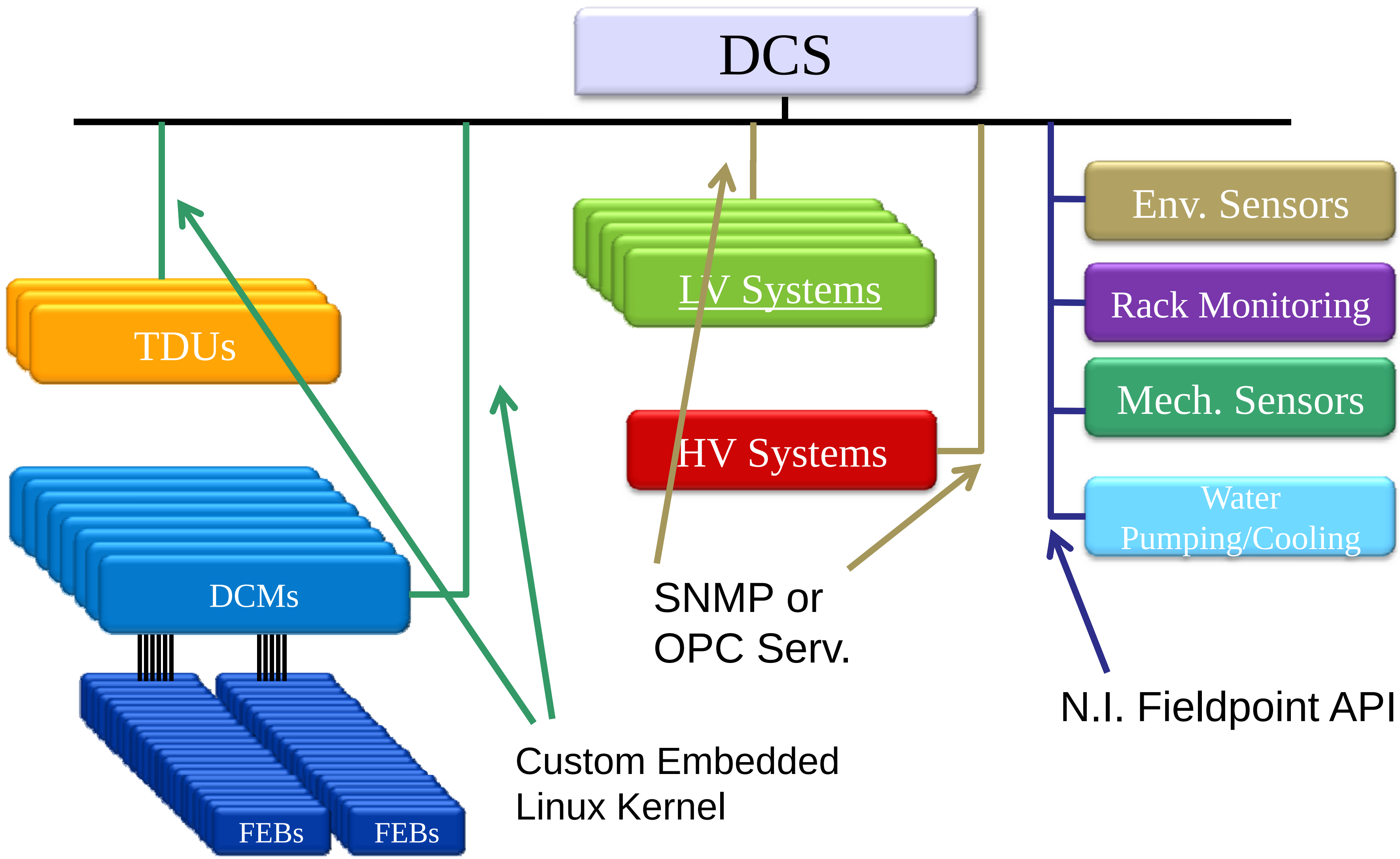
DCS Server Software

- Core system based off of EPICS developed at ANL
- Development started under base release R3.14 (Feb 2007 Rel.)
- Runs under Sci. Linux 4.x as well as Fedora Core
- Provides a Client/Server type architecture and communications protocol
- Operates under the concept of distributed Input/Output controllers with universal channel naming and access conventions
- Provides the majority of the transaction protocol, local state monitoring and channel history and access models
- Currently EPICS drivers exist for some of our hardware
  - Caen SY1527
  - N.I. Fieldpoints
  - Generic Ethernet device (base module)
- Need to develop drivers for
  - Wiener LV supplies
  - RS485 based devices (Standard OPCs will not work since we are using Linux OS)
  - DCMs (model off the basic Ethernet device)
  - FEBs (via DCM interface)
- Overhead and performance based on tests by ANL EPICS group, and from implementations at D0, were extrapolated to determine that NOvA will require 12 PCs to handle the monitoring/controls load (50k records per server).
- System is scalable. We can add additional IOCs to support new devices, or add additional servers to support higher loads.

Hardware Interfaces

- Interfaces follow vendor specific specs
  - Not possible to fully implement hardware/software interface layer till final specs and devices are prototyped/procurred
- Currently have samples for testing of:
  - Fieldpoint Readout Stations
  - Weiner PL508 & Mpod (ISEG HV)
  - Caen SY1527 HV and SY8800 LV
  - DCM Prototype + Firmware

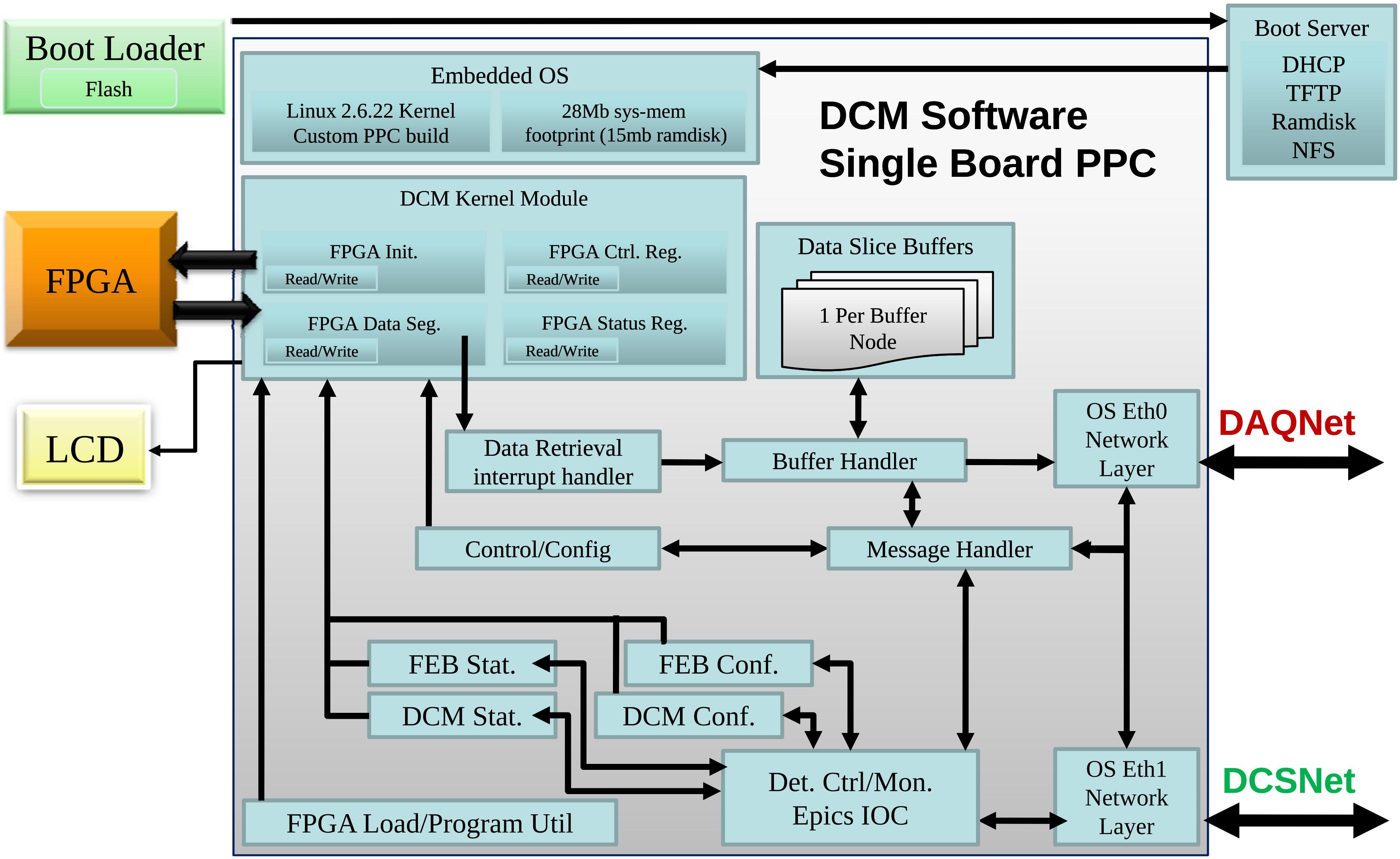
Control Protocols



DCM Embedded Software

- DCM runs a PowerPC single board computer
  - Standard chip, compilation rules exist
- Requires three types of Software development
  - Operating System
    - Custom Linux
    - Network boot, small memory foot print
  - Hardware driver
    - Custom Kernel module with support for FPGA and buses
  - DAQ clients
    - FPGA loader, DAQ inits, DAQ Readout, Buffer manager, Network dispatch
    - DCS client (readout/monitor/dispatch)

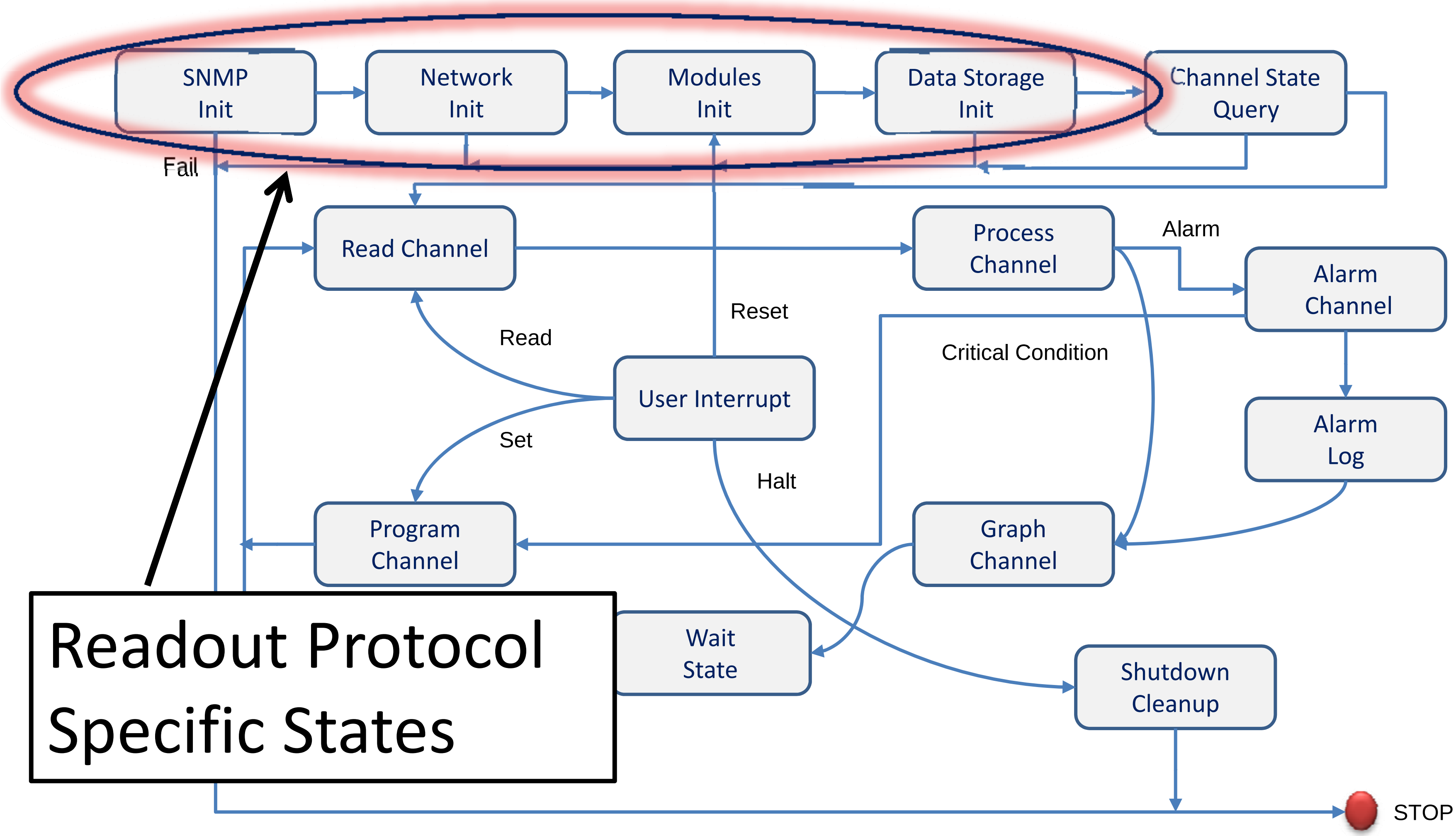
DCM Embedded Software



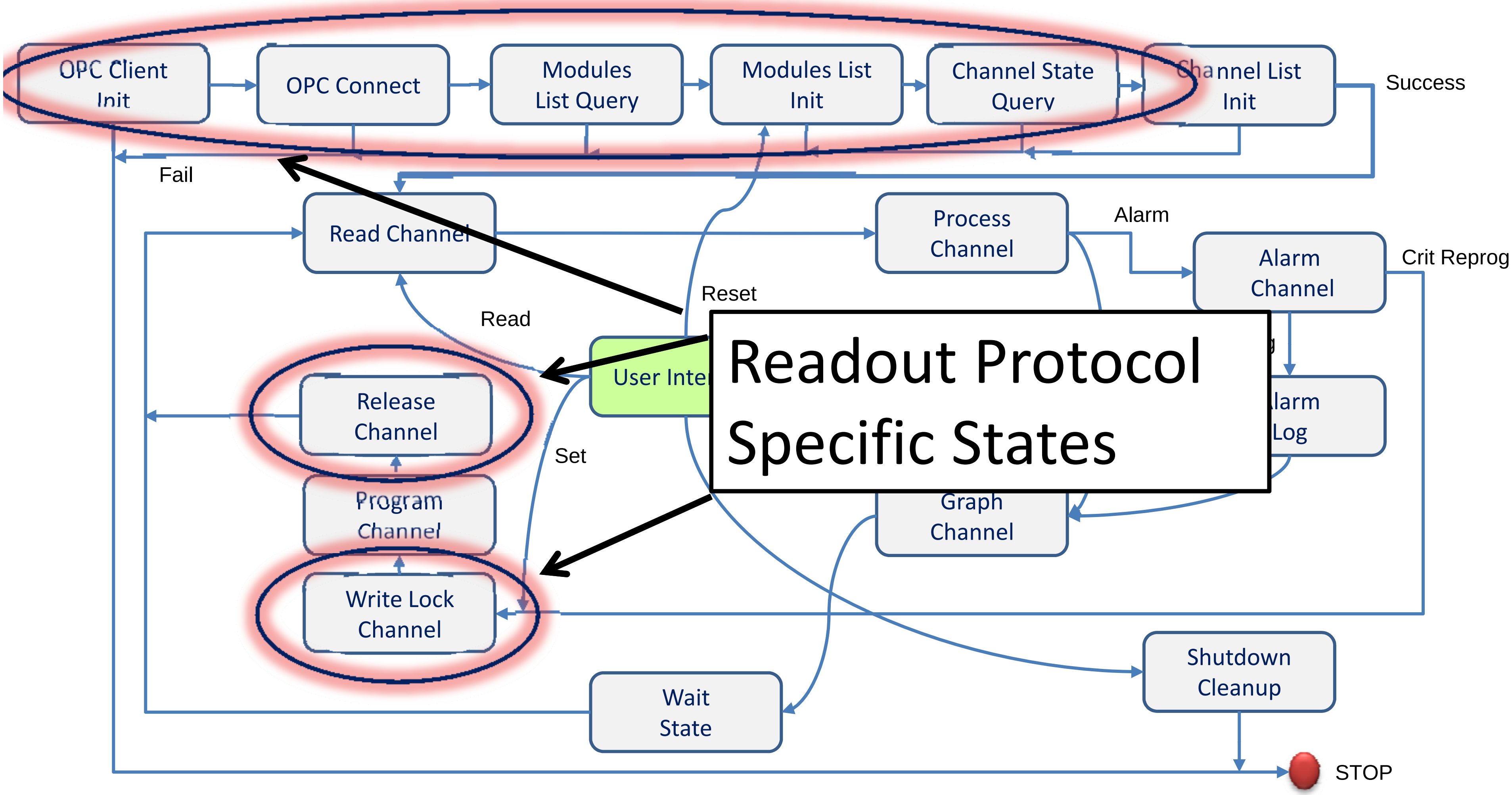
Preliminary Design

- Some systems have not been fully prototyped or demo models are not available
- We still design the preliminary readouts
  - Allows for modular design and integration with the EPIC IOC structure
  - Created state level diagrams for hardware/driver interaction
  - Implementation of the state machines can be tested and debugged using other control systems (e.g. Labview)
  - Allows for development of lightweight “test” clients which do not rely on full EPICs infrastructure

NOvA– Wiener MPOD-ISEG HV (SNMP)
Controls and Monitoring—State Transitions (IPND Software Design)



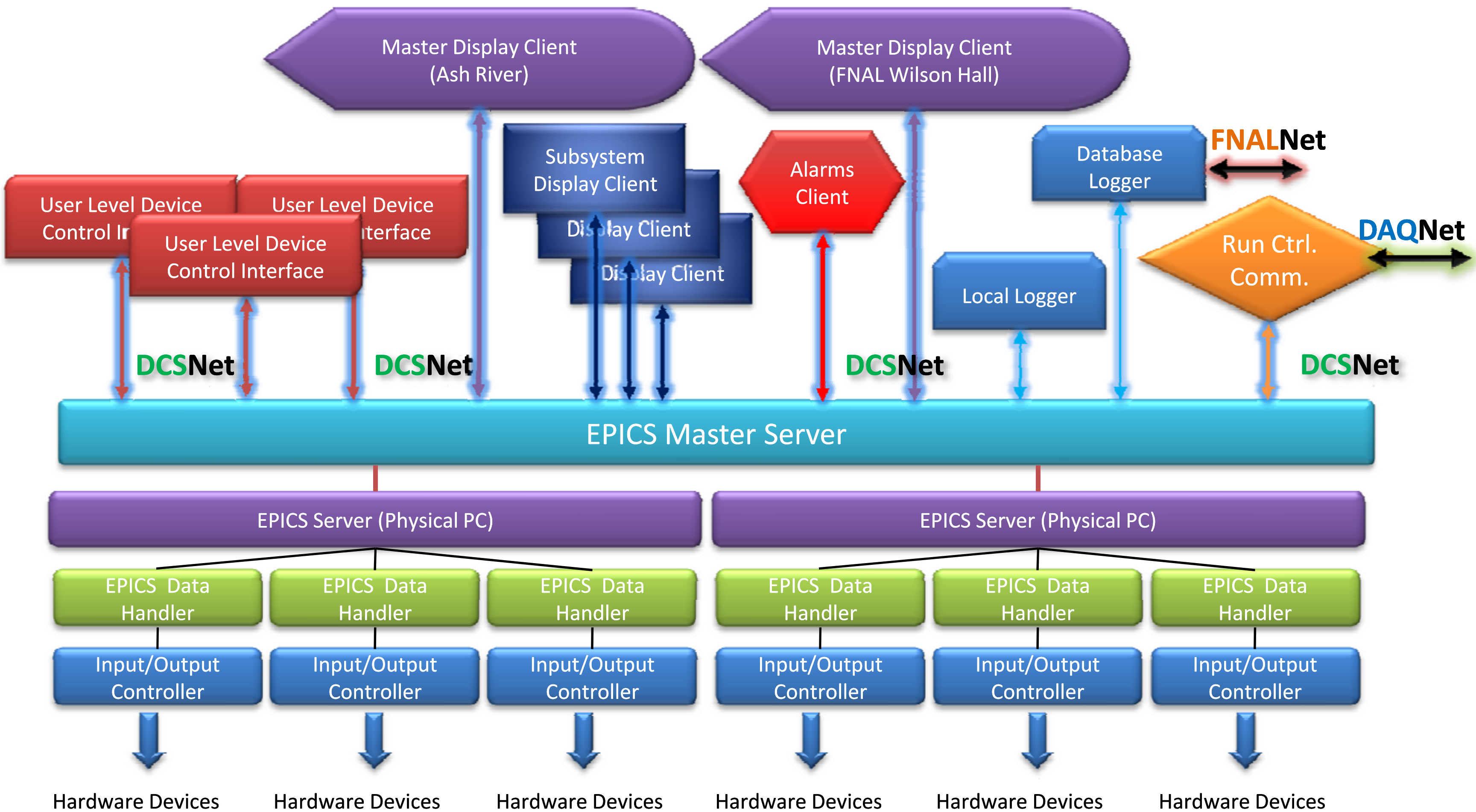
NOvA– CAEN SY1527 HV (OPC) Controls and Monitoring—State Transitions



Full EPICs design

- Modular hardware drivers allow us to integrate with the EPICs structure of
  - Input/Output Controllers
  - Channel Access Servers
- Clients are developed at the top of the design tree and see a generic interface and API for retrieving/monitoring data

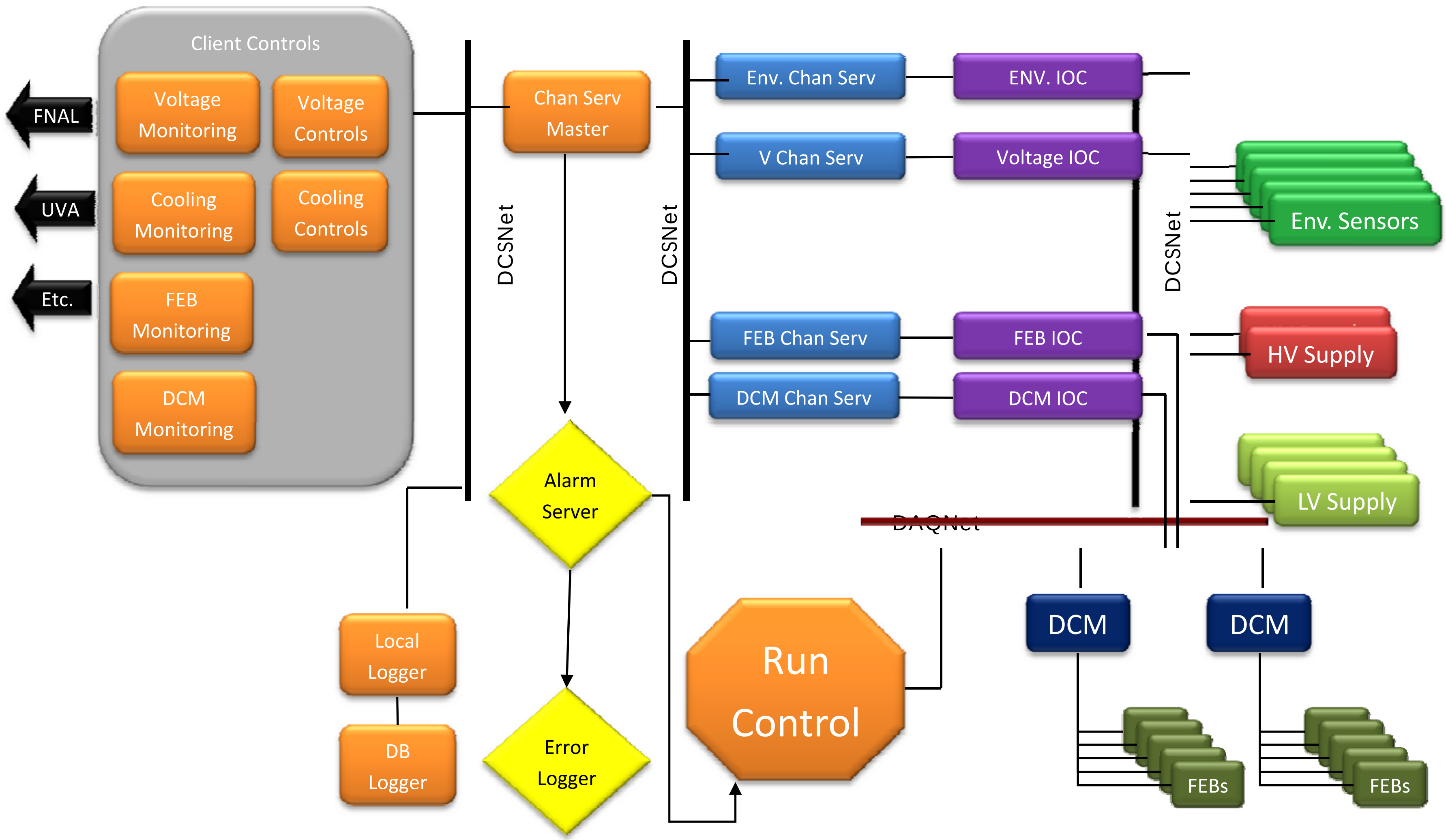
DCS Software Design Tree



DCS/DAQ Integration

- Det. Controls must interact with the main DAQ system
  - Provides access to program devices
  - Provides state/fault information
  - Acts as a general feed back channel for *hardware operations*
    - Not the same as the data quality monitor (DQM)
    - Not the same as the DAQ health monitor (DAQmon)

DCS Integration to DAQ & Logging



DCS/DAQ Integration

- Many subsystems share development paths between DCS and DAQ
- Primary system is the DCM
  - Kernel module developed to support both data stacks and monitoring data stacks
  - Client apps are being developed for both data path and dcs path in tandem
  - This inherits down to the FEB level since the DCM is the transport mechanism for talking to the FEBs

Detector Mass Based

BACKUP SLIDES

DCS Fieldpoint Readouts

- The detector hall monitoring, rack monitoring, the water cooling/pumping system controls, and misc sensors & systems are accessed through a common National Instruments Compact Fieldpoint station for each detector di-block.
- Stations backplanes are instrumented with 8 pods yielding 96-120 channels of controls or monitoring
- The break down of instrumentation types is:

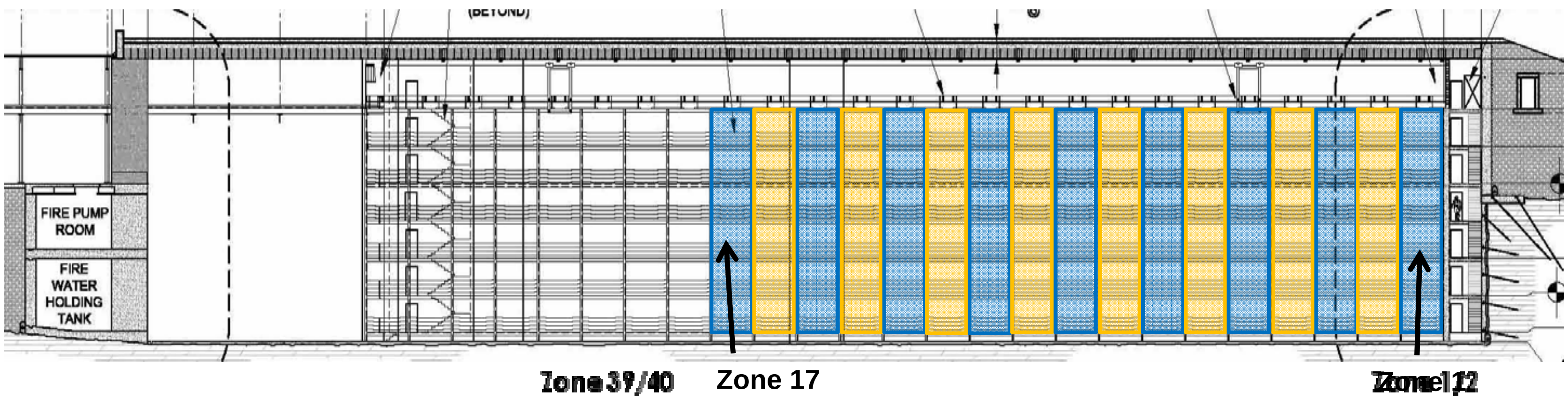
Pod Type	Slot	Units	Chan	Total Chan.	System
8 slot back plane	-	-	-	-	Base
Ethernet Control Interface module	0	1	-	-	Comm.
24 VDC, 120W rail mount supply (PS-5)	Rack	1	-	-	Sensor Power
3 wire RTD input 16bit	1	1	8	8	Cooling/Env.
10-30V Digital In/Out Module	2	1	8 (4/4)	8	Cooling
12-24V Digital Counter Module	3	1	8	8	Cooling
24V sinking Digital Input	4	1	32	32	Env/Cooling
16bit Analog Input (0-10V)	5	1	16	16	Env/Cooling
8 Chan Strain Gage Input	6/7	2	8	16	Block Stress
Open	8	1	32max	32	Misc
			Total:	120/96	

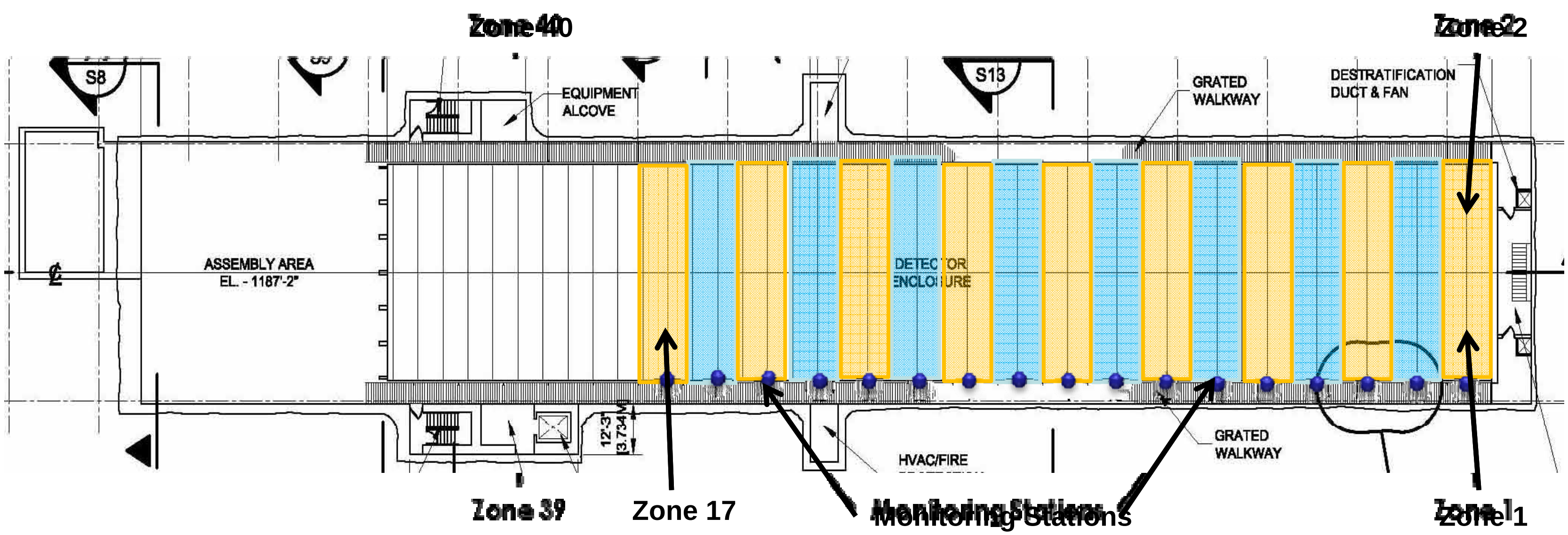
DCS Fieldpoint Readouts

- PS-5 24V power supply (external) used to power sensors, providing up to 120W for additional instrumentation on water cooling plant and optionally for timing/sync units.
- Provides integrated RS485 controller/bus
- Failsafe and disconnected operation permitted through onboard program cycle
  - Ensures network outage does not crash all readouts
  - Buffered readouts permit level-0 filtering of data and short term storage of some parameters
- Labview and OPC server options for local maintenance and hardware debugging.
- Product support lifetime 10+ years

Includes RS485/232 controllers

Monitoring Station Placement



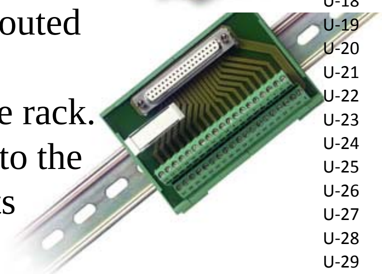


DCS Readout Rack

DCS readout hardware is housed in the di-block electronics racks

Simple sensor inputs to the field point stations are located on the front of the racks (integrated connection blocks)

Complex/Aux sensor inputs are routed through terminal block/board assemblies on the back side of the rack. Integration through DB37 cable to the backside inputs on the field points



RS485 devices are handled by Field point controller.

Readout of Wiener/CAEN devices is provided by connection to the local DCS network switch

Gbit Fiber (to Master)

Front

Position	Equipment
	DAQ Switch
U-3	DCS Switch
U-4	
U-5	Air Gap
U-6	Field Point
U-7	
U-8	
U-9	Exaust Fans
U-10	(3x 70mm muffin)
U-11	
U-12	Wiener PL508
U-13	
U-14	
U-15	
U-16	
U-17	
U-18	Exaust Fans
U-19	(3x 70mm muffin)
U-20	
U-21	Wiener PL508
U-22	
U-23	
U-24	
U-25	
U-26	
U-27	Exaust Fans
U-28	(3x 70mm muffin)
U-29	
U-30	Wiener PL508
U-31	
U-32	
U-33	
U-34	
U-35	
U-36	Exaust Fans
U-37	(3x 70mm muffin)
U-38	
U-39	Wiener PL508
U-40	
U-41	
U-42	
U-43	
U-44	

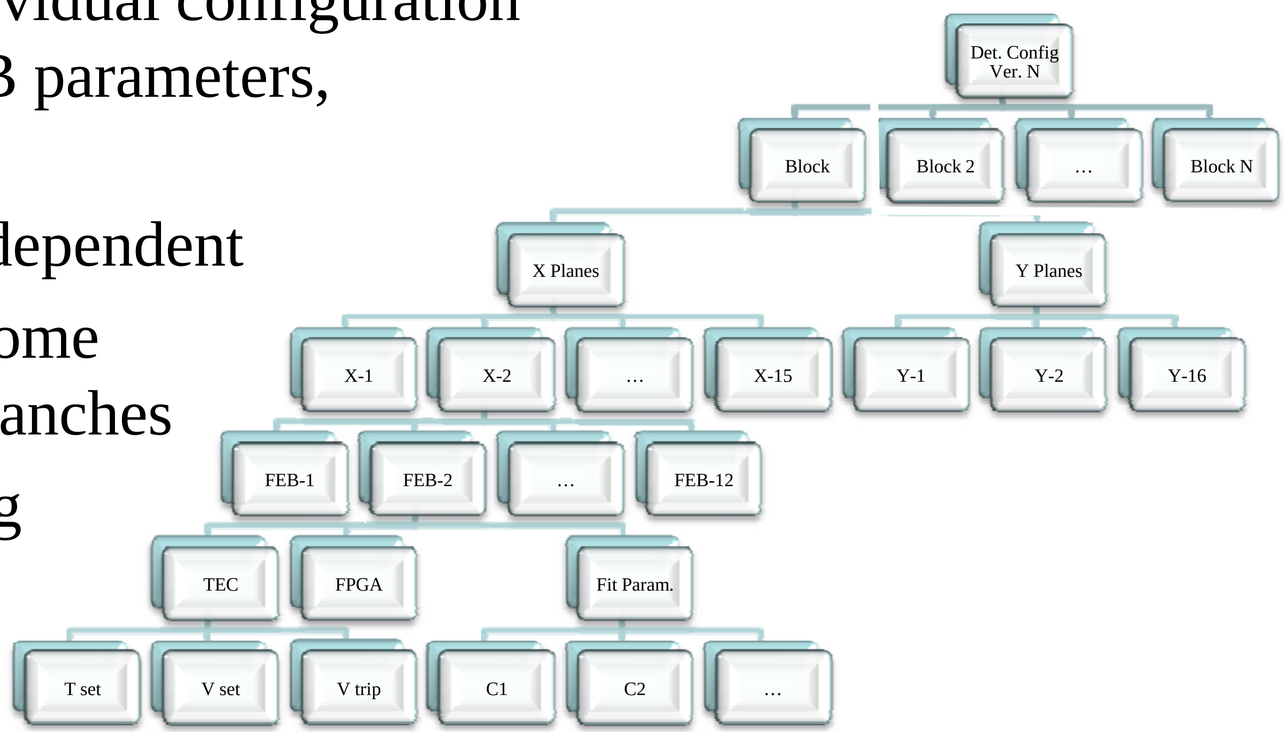
100Mb Ethernet

Back

Position	Equipment
U-1	OPEN
U-2	Aux. Terminal Block
U-3	DIN mounted
U-4	(DB37 Cable routing)
U-5	Air Gap
U-6	24V Power supplies
U-7	(Fieldpoint)
U-8	Power Distribution Block
U-9	(Fieldpoint)
U-10	Air Gap
U-11	Air Gap
U-12	Wiener PL508
U-13	
U-14	
U-15	
U-16	
U-17	
U-18	Air Gap
U-19	Air Gap
U-20	Air Gap
U-21	Wiener PL508
U-22	
U-23	
U-24	
U-25	
U-26	
U-27	Air Gap
U-28	Air Gap
U-29	Air Gap
U-30	Wiener PL508
U-31	
U-32	
U-33	
U-34	
U-35	
U-36	Air Gap
U-37	Air Gap
U-38	Air Gap
U-39	Wiener PL508
U-40	
U-41	
U-42	
U-43	
U-44	

DCS Configuration

- Need to be able to configure production and test configurations
- Typical controls configuration for a particular "detector" version can be broken down into a tree structure from the block level down to the individual configuration parameters (FEB parameters, voltages, etc…
- Branches are independent
- "Detectors" become collections of branches
- Easy partitioning and resource resolution



DCS Configuration

- This tree structure relates well to the Document Object Model (DOM) implementation of XML.
- Each detector object becomes a “node” with parent and child objects
- Values parameters become “leaf nodes”
- Full description of the detector component relationships

Node Definitions

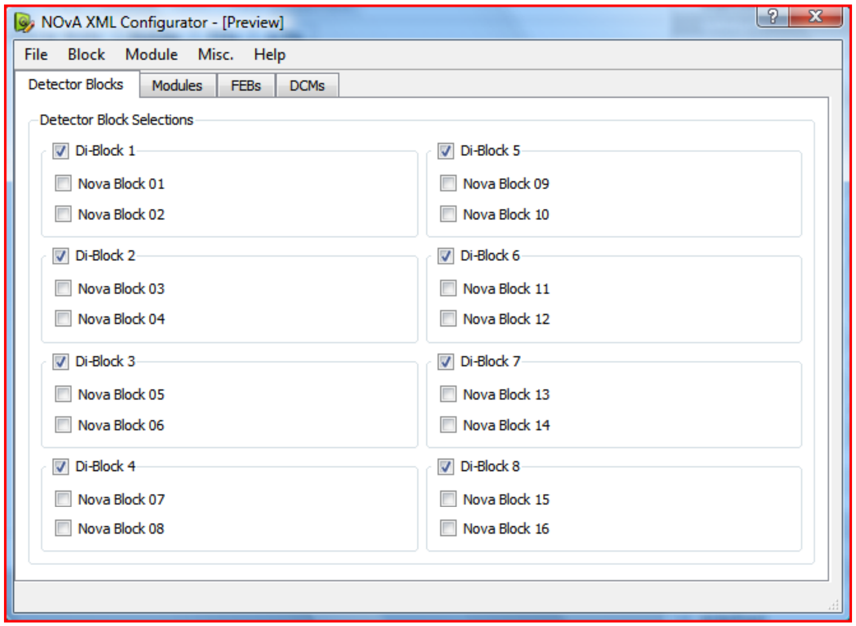
Leaf Values

```
<detector>
  <block blockname=“nova01”>
      <highvoltage  zone=“nova01-hv01”>450V</highvoltage>
      <highvoltage  zone=“nova01-hv02”>450V</highvoltage>
      ….
      <lowvoltage zone=“nova01-lv01” epicID=“lvmainframe-01”>
        <lv_chan epicID=“lv01-01” type=“3.3V”>
        <lv_vset epicID=“lv01-01-vset”>3.20V</lv_vset>
        <lv_ilimit epicID=“lv01-01-ilimit”>1.2A</lv_ilimit>
        </lv_chan>
      </lowvoltage>
      <module modulename=“nova01-x01” type=“X”>
        <FEB FEBname=“nova01-x01-f01” FEB_UNI_ID=“1”>
          <TEC tecname=“nova01-x01-t01”>
          <TEC_calibration>1.2</TEC_calibration>
          <TEC_setpoint>-15.0</TEC_setpoint>
          </TEC>
        </FEB>
      </module>
      <module modulename=“nova01-y02” type=“Y”>
      ….
      </module>
  </block>
  <block blockname=“nova02”>
  </block>
…
</detector>
```

DOM/XML Advantages

- Provides a natural hierarchy with inheritance of parent/child relationships that mirrors the physical detector layout
  - Allows for identification and configuration of a “detector region” instead of a discrete set of channels
  - Allows for alarm association and correlations
- Allows us to embed the EPICs channel information into the description file so that it’s abstracted from the operator
- Parsing is done through nodes lists allowing us to extract subsets of data (i.e. all low voltage channels in Block02, or all detector active TECs)
  - Allows client applications to use the general config file instead of requiring each application to have a separate list of components
- Human readable/editable configuration files
  - But with a configuration interface for generating base configs
- Relies on standard APIs for file parsing/writing
- Has natural translation modules into SQL tables
  - Ease database integration
  - Provides a natural mapping from central DB to runtime config
- Industry standard for document markup

Configurator



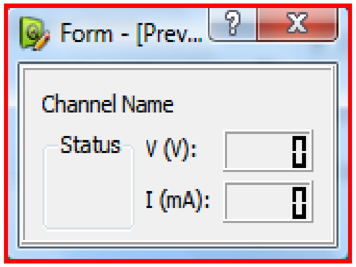
- Loads baseline values
  - Either from central DB
  - Or XML file
- Allows for detector configuration on any level of granularity
  - Block, Module, FEB
- Output is DOM style XML to file or back to DB
- Used to create client UI's on the fly with only the relevant detector elements
- All UI's use the same config file and only parse out what they need (i.e. HV or LV channels, DCM parameters etc…)

Client Building

- DCS clients are “built” at run time by parsing the config file and instantiating control widgets into display arrays
- Involves no static assumptions about the current detector
  - Allows for detector growth
  - Allows for small “test” clients
  - Allows multiple clients to run in parallel
- Channel maps are automatically propagated to clients without rebuilds
- EPICs layers are completely abstracted

- Requires a central resource manager to resolve resource allocation conflicts

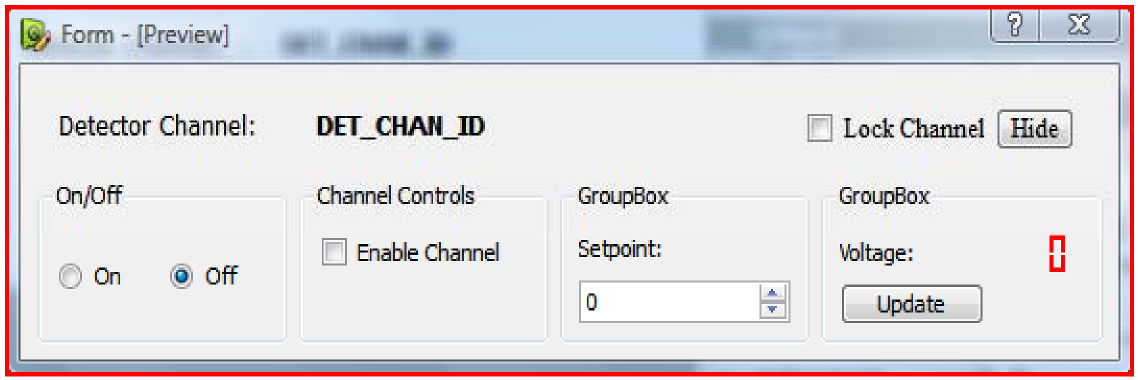
Widget Examples: Voltage

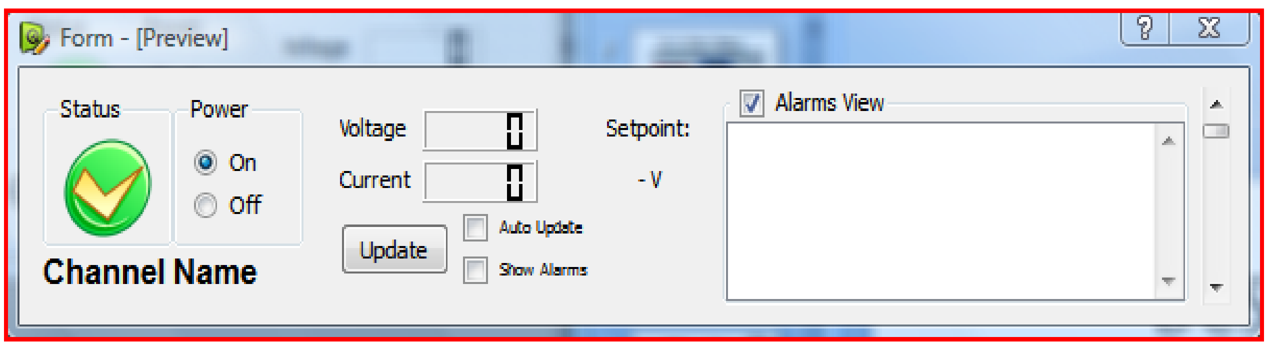


• Compact Voltage display widget which shows only voltage and current for a generic channel ("Channel Name" is replaced on instance init, with the EPICS channel name)

•Simple control widget for a single voltage channel. The widget allows for the on/off toggle, channel enable/disable, and modification of the setpoint.

• An update field is included with a readback voltage and update.

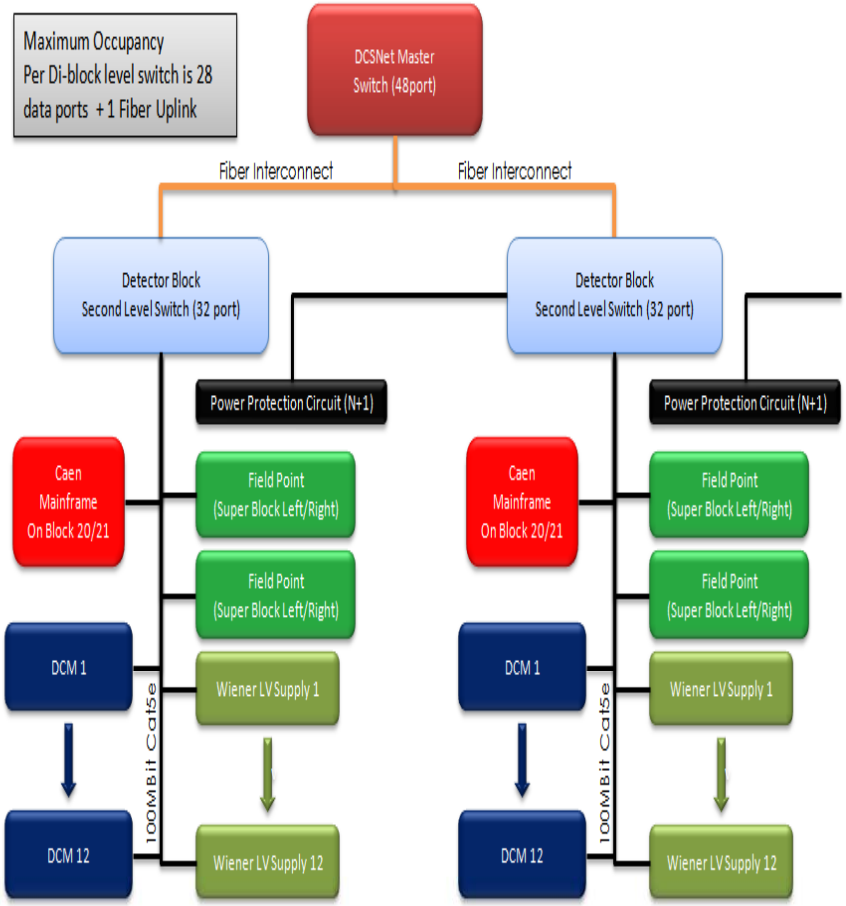




• The "detailed" voltage monitoring widget which includes a nice graphic status display, power indicator (not a control), a readback for both voltage and current, and a window showing a simple alarm history

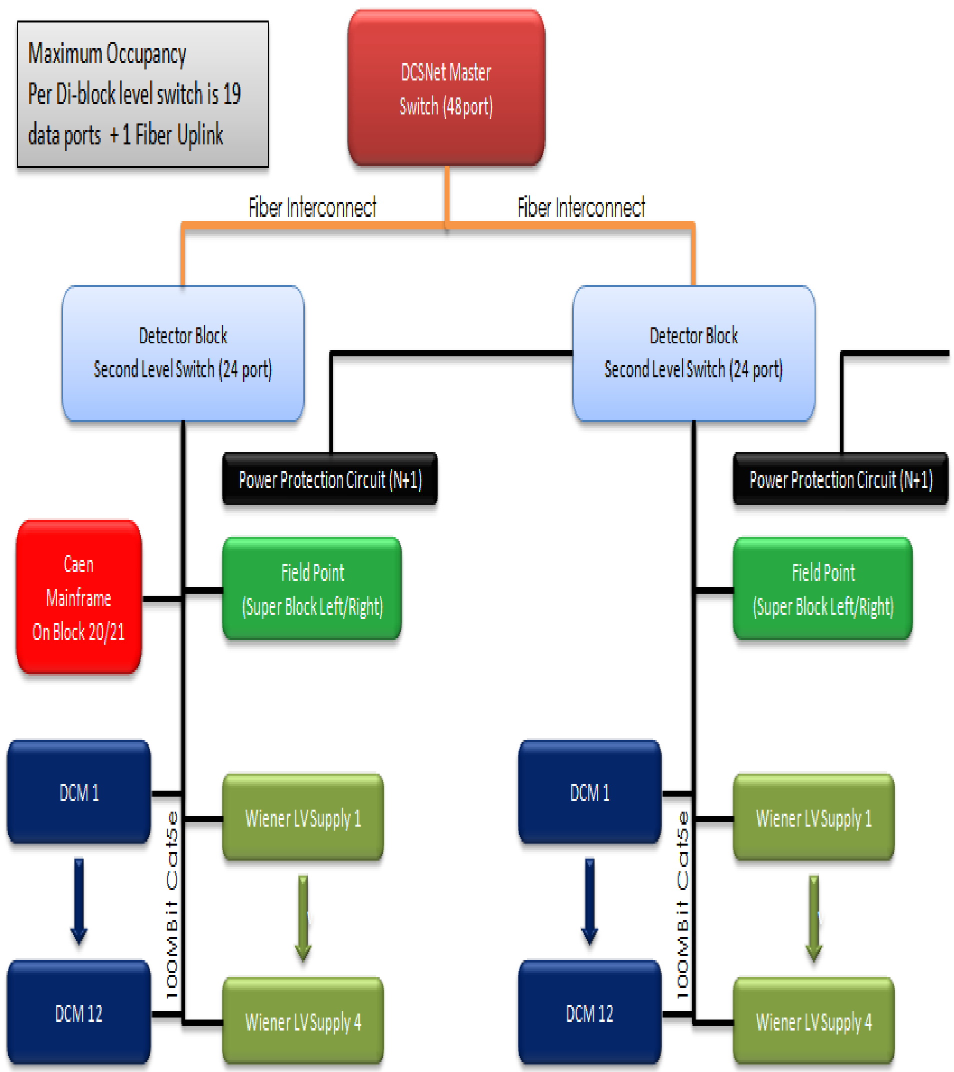
Second Level Switch Occupancy

- 20 second level switches
- Gbit bandwidth per switch
- Fiber uplink to master switch in electronics room
- Aligned by detector di-blocks on detector Left/Right.
- Mounted in catwalk elec. racks
- Each switch handles minimum load of 12 Wiener LV supplies, 2 Fieldpoint control stations, rack power control (for rack N+1)
- Full DCS traffic partitioning include the DCM's 2nd Ethernet port on the switch
- Total Occupancy: 28 ports



Second Level Switch Occupancy

- 17 second level switches
- Gbit bandwidth per switch
- Fiber uplink to master switch in electronics room
- Aligned by detector di-blocks on.
- Mounted in catwalk elec. racks
- Each switch handles minimum load of 4 Wiener LV supplies, 1 Fieldpoint control stations, rack power control (for rack N+1)
- Full DCS traffic partitioning include the DCM's 2nd Ethernet port on the switch
- Total Occupancy: 19 ports
  - Means we need a 24port switch



Scope: Far Detector Controls (20kT)

Subsystem	Type	Units	Channels	Total Channels	Interface
High Voltage	Caen SY1527	2	96	192(114 used)	Ethernet
Low Voltage	Wiener	76	6	456	Ethernet
Water Cooling	Neslabs M150	38	10	380	Ethernet/RS485
Data Acq.	Custom DCM	228	62(64)	14,136	Ethernet
Det. Envir.	N.I. Fieldpoint	38	128	4,864	Ethernet
Front End	Custom FEB	14136	32	452,352	8B/10B DCM
Timing/Sync	Custom TDB	11	4	44	Ethernet
Misc. & Safety	Misc. Sensors	200	1-10	2000	Misc.
			Total	22,072 474,424 w/FEBs	

- The detector controls system is required to have access to over 15,000 physical devices and 475,000 independent programmable channels
- Controls interface must present a uniform method of accessing all the devices regardless of controls interface
- The system must be modular, scalable and support partitioning into production, installation, and calibration/commissioning variants to accommodate the “physics during build” model.

Scope: Far Detector Monitoring (20kT)

System	Parameters/Chan	Channels	Total Monitoring
High Voltage	6	80	486
Low Voltage	4	456	1,824
Water Cooling	2	380	760
DCM	24	228	5,472
Det. Env	128	38	4,864
Front End	20-25	14136	353,400
Timing/Sync	10-15	11	165
Misc. & Safety	1-2	2000	4000
		TOTAL	17,571 370,971 w/FEBs

- The detector Monitoring system is required to monitor and alarm on over 370,000 system critical run time parameters.
- The monitoring frequency must be variable from ~1Hz for pushed data streams to once per minute (or 5minutes) for slower polled readouts.
- Client initiated monitoring requests can cause burst spikes in the DCS bandwidth with transfers on the order of 400-600mb per full monitoring dump with protocol and network overhead