LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

| | |
|---|---|
| **Document Type** | **LIGO-T950120-01 - CCD** 1/31/96 |

# CDS Control and Monitoring Conceptual Design

R. Bork, D. Barker

*Distribution of this draft:*

This is an internal working note
of the LIGO Project.

**California Institute of Technology**
**LIGO Project - MS 51-33**
**Pasadena CA 91125**
Phone (818) 395-2129
Fax (818) 304-9834
E-mail: info@ligo.caltech.edu

**Massachusetts Institute of Technology**
**LIGO Project - MS 20B-145**
**Cambridge, MA 01239**
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

WWW: http://www.ligo.caltech.edu/

# TABLE OF CONTENTS

# TABLE OF CONTENTS

# TABLE OF CONTENTS

# 1 INTRODUCTION

## 1.1. Purpose

The purpose of this document is to describe the overall design of the LIGO Control and Data Systems (CDS) control and monitoring system.

## 1.2. Scope

CDS for LIGO has been divided, for design purposes, into two major components: Control and Monitoring and Data Acquisition. The Data Acquisition system is described in other documentation, namely the requirements in LIGO T960009-C and conceptual design in LIGO T960010-C.

This document describes a conceptual design for the infrastructure to be employed in the control and monitoring system, which includes:

- LIGO control room systems, including operator consoles.
- Computer networking systems
- Timing systems to accurately timestamp LIGO data, both for control and monitoring and for LIGO data acquisition.
- Standard Input/Output (I/O) systems to be used to interface to equipment to be controlled/monitored (referred to, within this document, as front end systems).
- High level control and monitoring application and development software.

This is the highest level of CDS control and monitoring design documents, a direct outcome of the CDS Control and Monitoring Requirements Document (LIGO T950054-C), as shown in Figure 1: LIGO Requirement Specification Tree. Other CDS design documents cover specific hardware and software implementations applied to the various LIGO interferometer, vacuum and physics enviroment control subsystems.

## 1.3. Document Overview

This document represents the conceptual design and will be "frozen" at the time of the CDS Global Design Requirements Review (DRR). This document will then be copied into and further detailed in a Preliminary Design Document (PDD), which (less the final detailed installation drawings), becomes the LIGO CDS design which will be implemented. An updated PDD and all installation plans and drawings will then become the final design package.

This document details a design which depicts what would be installed in the system if the system were to be deployed in the near term with today's technology. Since CDS involves a large amount of computer and related equipment, it is anticipated that these technologies will advance over the next two years before major purchases are made. Therefore, specific equipment selections shown in this document will change over the design phase, but the general concepts should still be valid.

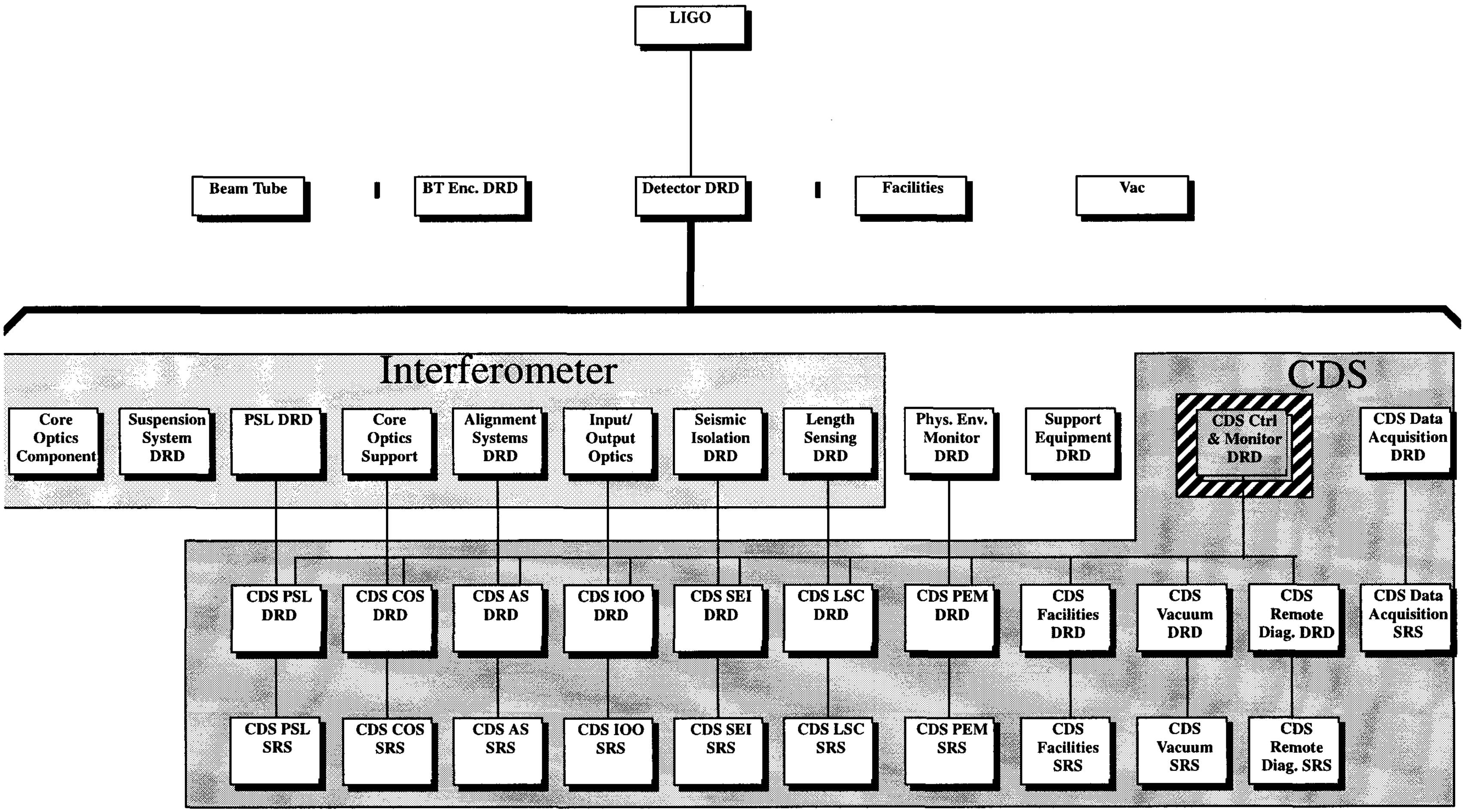



**Figure 1: LIGO Requirement Specification Tree**

## 1.4. Definitions

## 1.5. Acronyms

- ALM - EPICS Alarm Manager
- API - Application Programmer's Interface
- APS - Advanced Photon Source (Argonne National Lab)
- AR - EPICS data archiver
- ARR - EPICS archive retrieval tool
- ATM - Asynchronous Transfer Mode
- BURT - BackUp and Restore Tool
- CA - Channel Access
- CDS - Control and Data System
- CEBAF - Continuous Electron Beam Accelerator Facility
- CIM - Computer Integration and Manufacturing
- CPU - Central Processing Unit
- DCS - Distributed Control System
- DRD - Design Requirements Document
- EPICS - Experimental Physics and Industrial Control System
- EZCA - Easy Channel Access
- FC - Fiber Channel
- FCMS - Facility Control and Monitoring System
- FCR - Facility Control Room
- FDDI - Fiber Distributed Data Interface
- GPS - Global Positioning System
- GUI - Graphical User Interface
- HMI - Human Machine Interface
- HPPI - High Performance Parallel Interface
- Hz - Hertz
- IEEE - Institute of Electronic and Electrical Engineering
- IFO - Interferometer
- I/O - Input/Output
- IP - Internet Protocol
- ISO - International Standards Organization
- IXS - Information eXchange Services
- LAN - Local Area Network
- LANL - Los Alamos National Laboratory
- LIGO - Laser Interferometer Gravity wave Observatory
- LVEA - Laser and Vacuum Equipment Area
- MAC - Media Access Layer
- MEDM - Motif Epics Display Manager
- MHz - Mega Hertz
- NASA - National Aeronautics and Space Administration
- NSAP - Network Service Access Port
- OSB - Operations Support Building
- PDD - Preliminary Design Document

- PSL - PreStabilized Laser
- TBD - To Be Determined
- TCP - Transport Control Protocol
- UDP - User Datagram Protocol
- UPS - Uninterruptable Power Supplies
- VAC - Volts Alternating Current
- VDC - Volts Direct Current
- VME - Versa Modular Eurocard
- VXI - VME eXtensions for Instrumentation

## 1.6. Applicable Documents

### 1.6.1. LIGO Documents

CDS Control and Monitoring DRD LIGO T950054-C

CDS Data Acquisition System DRD LIGO T960009-C

CDS Data Acquisition System Conceptual Design LIGOT960010-C

CDS Vacuum Cabling and Feed Through DRD LIGO T950095-C

### 1.6.2. LIGO Drawings

### 1.6.3. Non-LIGO Documents

### 1.6.4. Non-LIGO Drawings

# 2 SYSTEM OVERVIEW

The control and monitoring portion of CDS is designed as a Distributed Control System (DCS). A basic block diagram of the hardware arrangement is shown in Figure 2: CDS Overview.

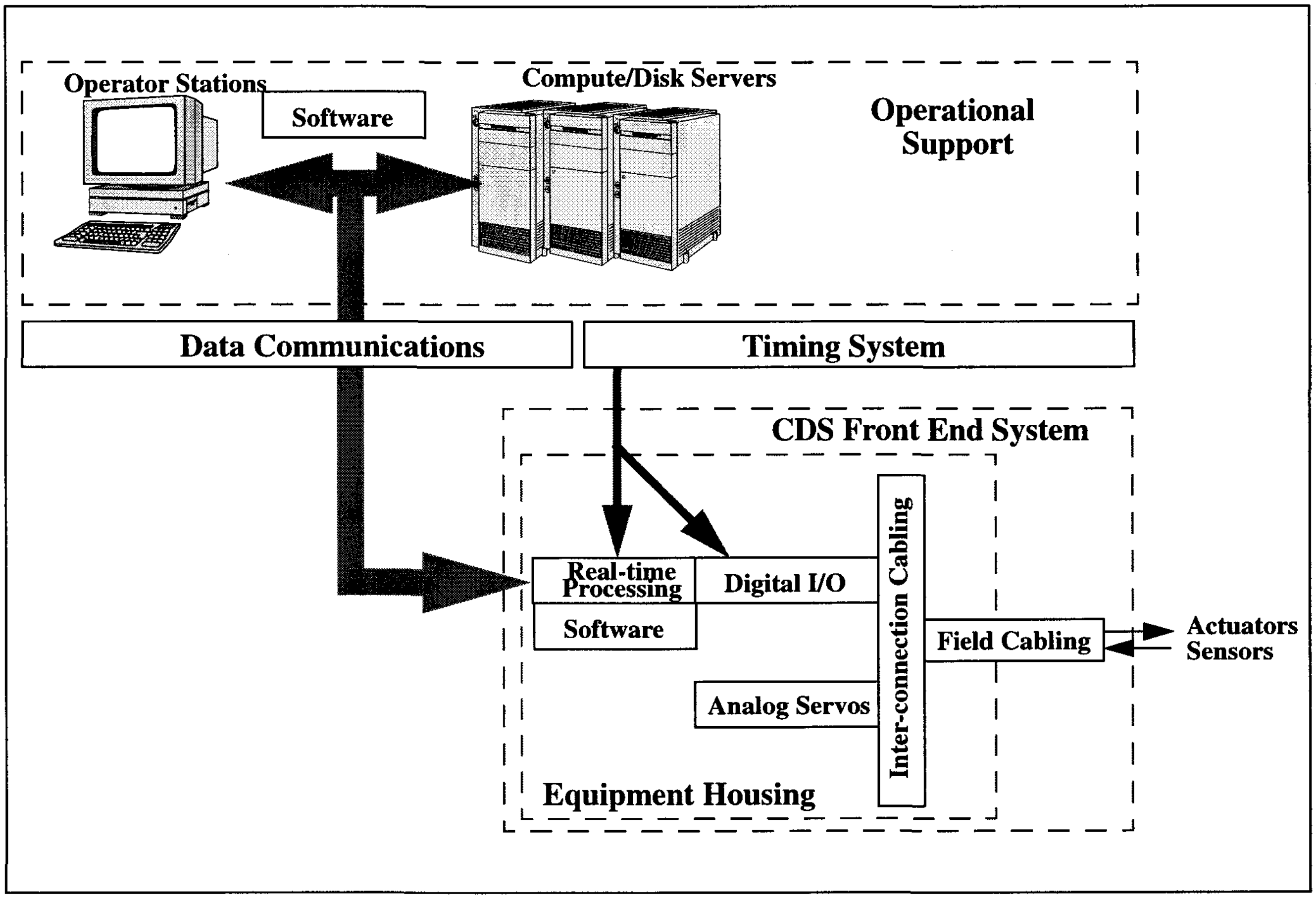



**Figure 2: CDS Overview**

# 3 FRONT END SYSTEMS

Front end systems are the field units which connect the CDS to the equipment to be controlled and/or monitored. The following sections describe the standard equipment to be employed in these systems.

## 3.1. Front End Processor / I/O Bus

### 3.1.1. Design

The CDS will employ Versa Modular Eurocard (VME) standard crates and modules for front end I/O and processing. The VME crates meet LIGO CDS standard TBD. VME eXtensions for Instrumentation (VXI) systems will also be used in limited applications.

As shown in the standard rack layout sketch, a VME crate will exist in each CDS rack pair (on average). The VME crate will feature a "split" backplane. Slots 0-10 will house the control and monitoring processor(s) and I/O modules. Slot 11 will not have backplane connections. Slots 12-21 will contain the CDS DAQ processor(s) and I/O modules. This is done to save on crate costs, as initial estimates indicate that separate crates would underutilize the slot space for control and monitoring and data acquisition, and therefore functions can be combined into the same crate. Estimates of VME bus usage, however, indicate that the two functions require separate backbones.

### 3.1.2. Design Analysis

Any number of I/O and/or processor busses are available today which would meet LIGO needs. Some of the reasons VME is chosen over other buses are:

1. Commercial standard, with support from numerous vendors.
2. Relatively high performance / low cost.
3. Open standard architecture allowing for custom module designs.
4. High versatility. Many VME based products available, from standard I/O, to processors, to DSP modules, which allows a great flexibility for CDS design and future upgrades.

Due to its higher cost, VXI systems will be employed in a capacity limited to special test equipment setups where VXI instrumentation modules are available. This will be primarily in the realm of CDS Remote Diagnostics (described in separate documentation).

## 3.2. Analog Servo and Signal Conditioning Hardware

Analog servo and signal conditioning units will be designed and manufactured as 6U Eurocards. These will be installed into LIGO standard Eurocard cages similar to the LIGO VME crates. However, the backplanes for these crates will only provide power connections, not digital lines. Such a crate is shown in Figure 3: CDS Eurocard Cage.

This housing has the same appearance as the CDS VME crates. However, as seen from the top view, two backplanes are installed, allowing modules to be inserted in both the front and back of the unit. Sensitive analog circuit boards would be installed in the front, with modules containing high voltage and power circuitry installed into the rear slots. Space is provided in the center between the backplanes to allow access for interface cabling.

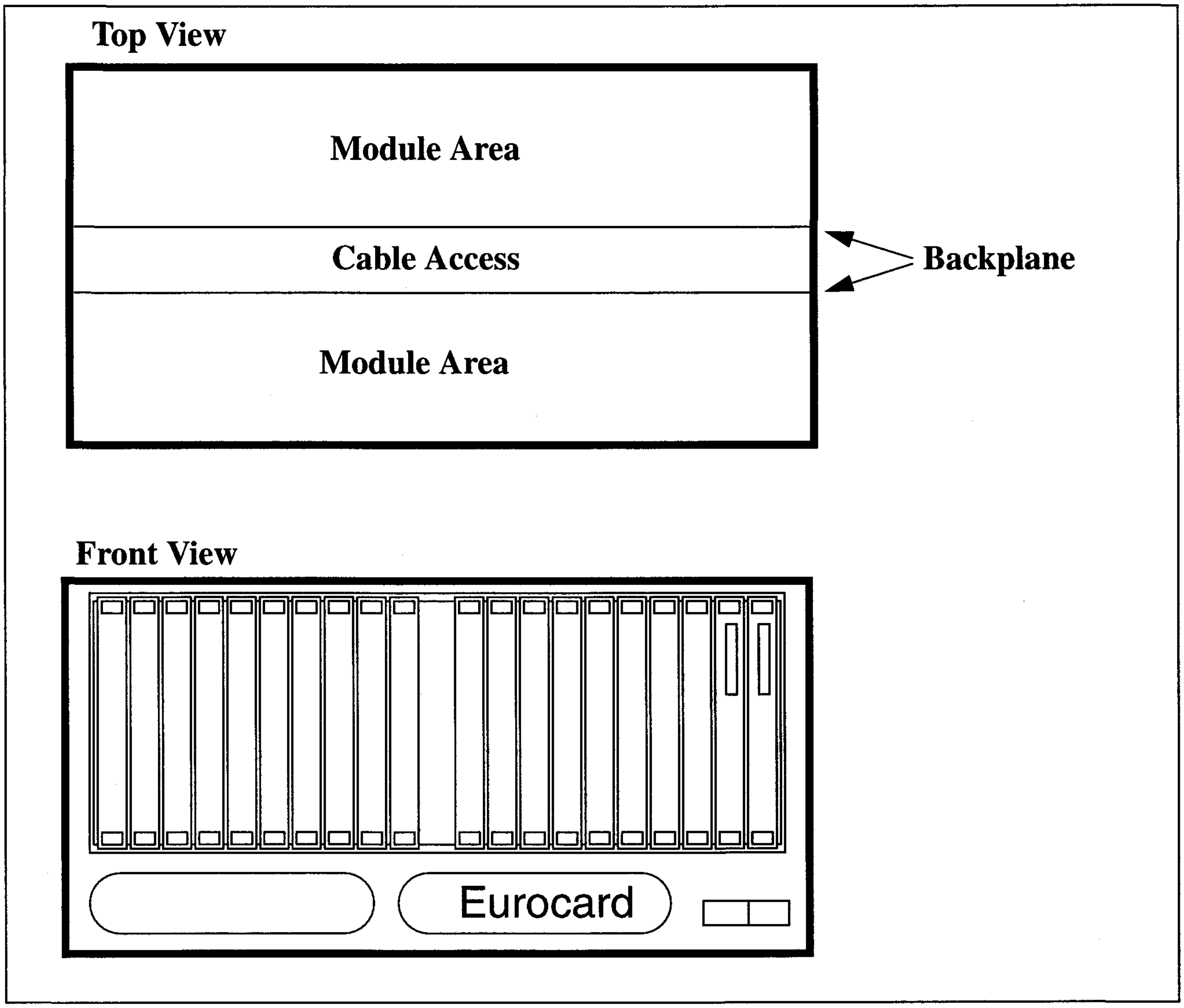
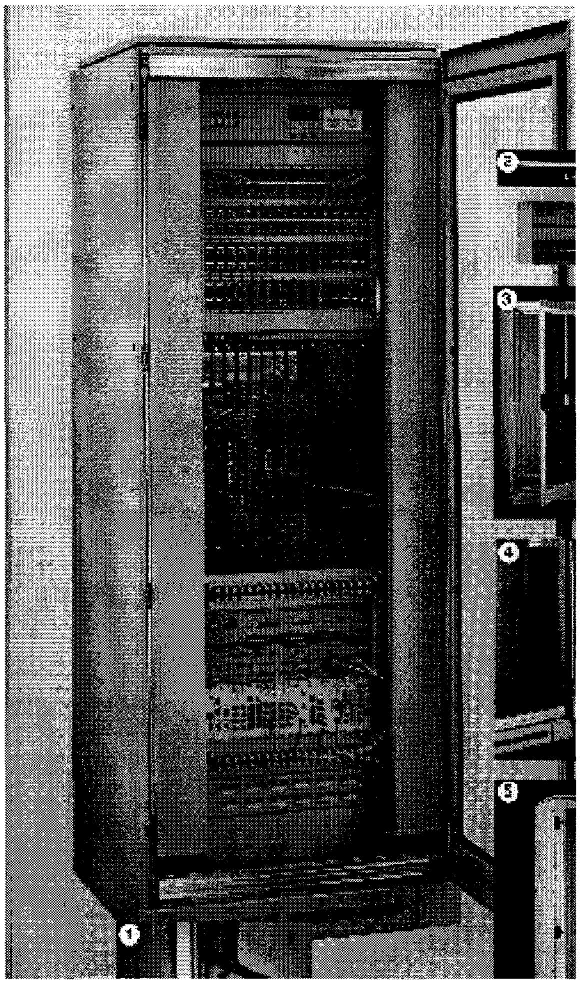



**Figure 3: CDS Eurocard Cage**

# 3.3. Equipment Housing

## 3.3.1. Design Specifications

CDS front end systems will be contained in standard 19" rack mounting systems. The standard rack and specifications are shown in Figure 4: CDS Rack Standard.



- Aluminum and sheet steel
- Dimensions: 27.6" x 31.5" x 84"
- Doors: Front/Rear/Side
- Side cable feed base
- 41U installation area (71.75")
- Load Capacity: 225 lbs.
- Power dissipation: 700W max.

**Figure 4: CDS Rack Standard**

## 3.3.2. Typical Layout

A typical front end rack layout is shown in Figure 5: CDS Standard Front End Rack Assembly (LVEA). This rack contains:

1. A 1U top panel (Service Panel), which includes:
   - Panel breaker(s) for rack power
   - 10baseT connector which provides an ethernet connection to the CDS networks. This allows for connection of a laptop PC for local operation/maintenance.

- Phone jack

2. Two 1U 24VDC power supplies. +/-24VDC will be the CDS standard for binary I/O operation, such as relays, switches, contacts, etc. These are also the standard supplies for CDS analog servo circuit boards.
3. VME and Eurocard crates
4. Wiring cross connect systems
5. Phone

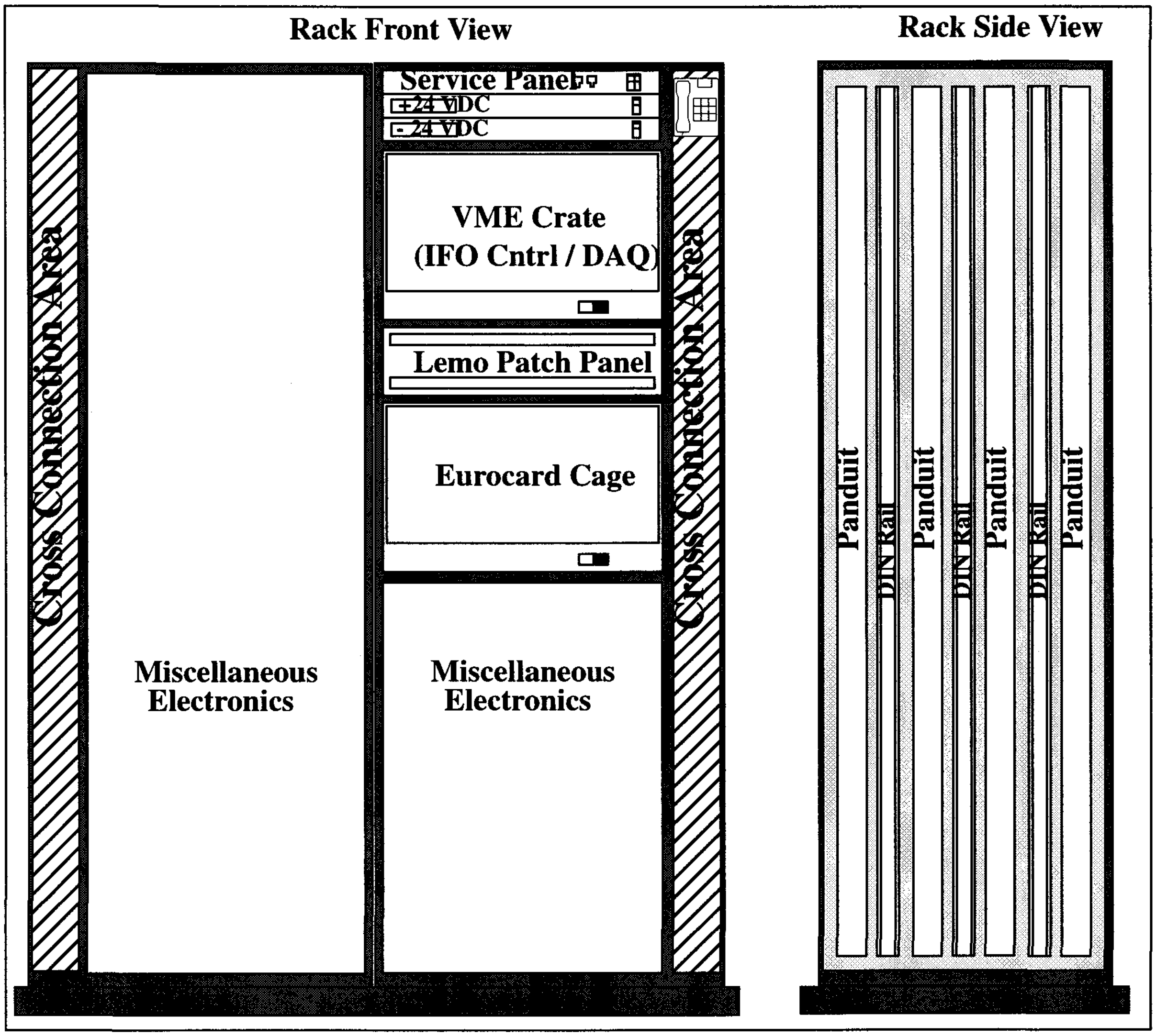



**Figure 5: CDS Standard Front End Rack Assembly (LVEA)**

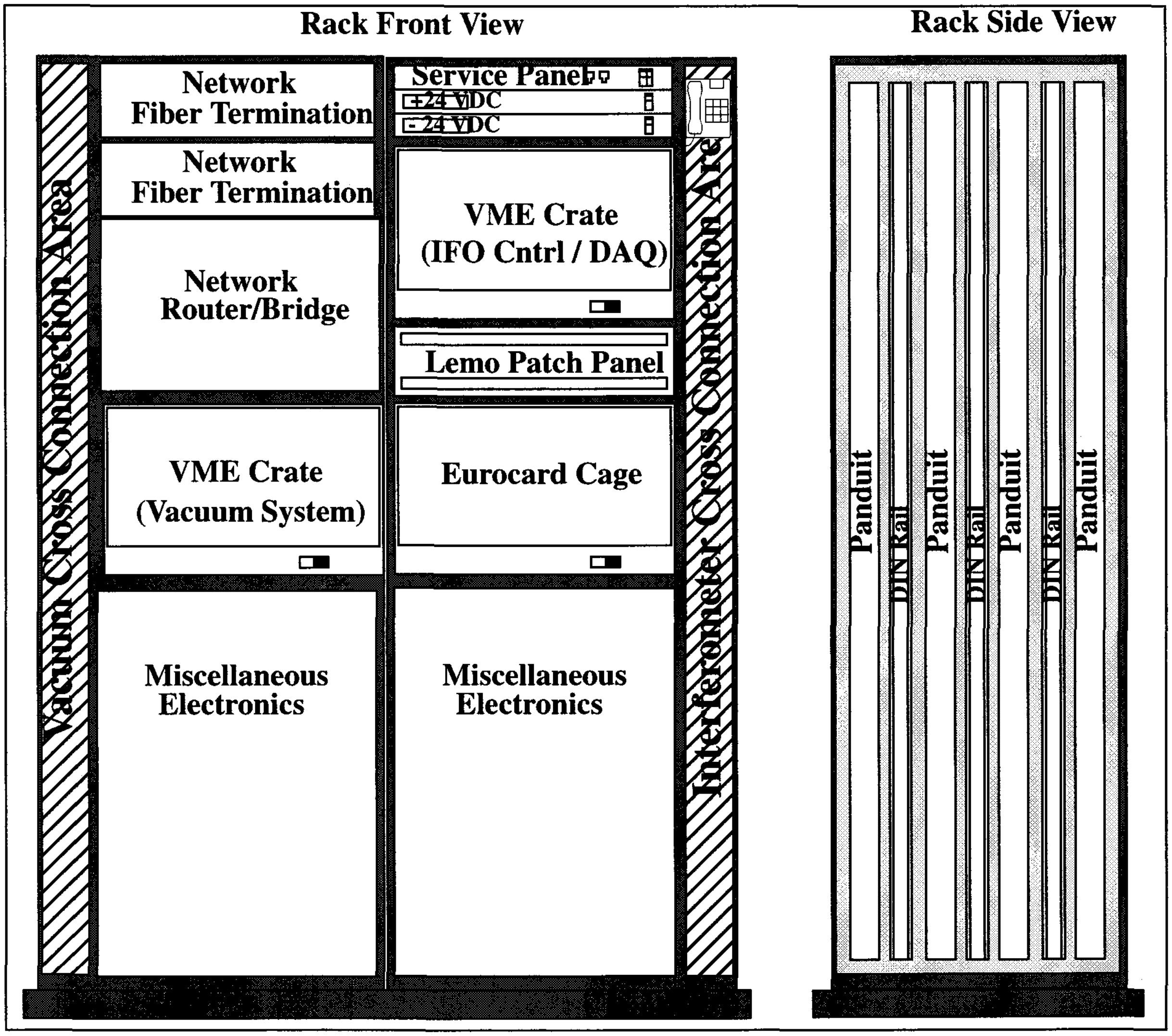



**Figure 6: Standard Rack Layout (Mid and End Stations)**

## 3.4. Inter-connection wiring

### 3.4.1. Design

Three general schemes will be used to interconnect CDS modules and CDS to external equipment:

1. Coax wiring through a Lemo patch panel.
2. Interconnect wiring through DIN rail mounted discreet or mass termination blocks.
3. Signal cables designated as "critical" due to signal levels, noise levels, allowed lengths, etc., will be run directly from the CDS interface module(s) to the equipment involved.

As shown in the previous rack layout sketch, a Lemo patch panel will exist between the control and monitoring and DAQ crates. This will be the primary interface point to control signals which must be acquired and archived by the DAQ system. This panel will also provide the connection point for signals interfaced via coaxial cable to field equipment and provide some standard test point signals for o'scope connection.

Most signals will be interconnected through the DIN equipment housed in one side of the rack. This method is shown in Figure 7: CDS Cable Interconnect System. This cross connect area contains alternating Panduit (or equivalent) tray and DIN rails. The DIN rails will be used to mount Phoenix (or equivalent) mass termination blocks for connection of multi-conductor ribbon cable from VME I/O modules to field devices and single point termination blocks for distribution of power and similar types of signals. The standard mass termination cable is a twisted, shielded, round ribbon type, with flat ribbon breakouts at 2 meter intervals for the standard ribbon connectors. The routing of cable from the VME and other front panel electronics to the cross connect area will be via punchouts in the Panduit areas through the rack side wall into the main rack area.

Cable routing into/out of the racks will be via the bottom of the rack. In LIGO equipment buildings, cabling will be through a toe base into a floor-mounted cable tray. Those racks located within the control areas of the OSB will be via openings in the false floors.

### 3.4.2. Design Analysis

For coaxial cable connections, Lemo connectors will be used primarily due to their more compact size, making them easier to fit on VME modules (most commercial VME modules with coaxial connections use Lemo).

Several methods for the majority of cable interconnects were studied. The DIN rail mass and single termination method was chosen for its high availability of components, standard commercial usage, and clean layout capabilities. This method was successfully employed on the LIGO PSL prototype.

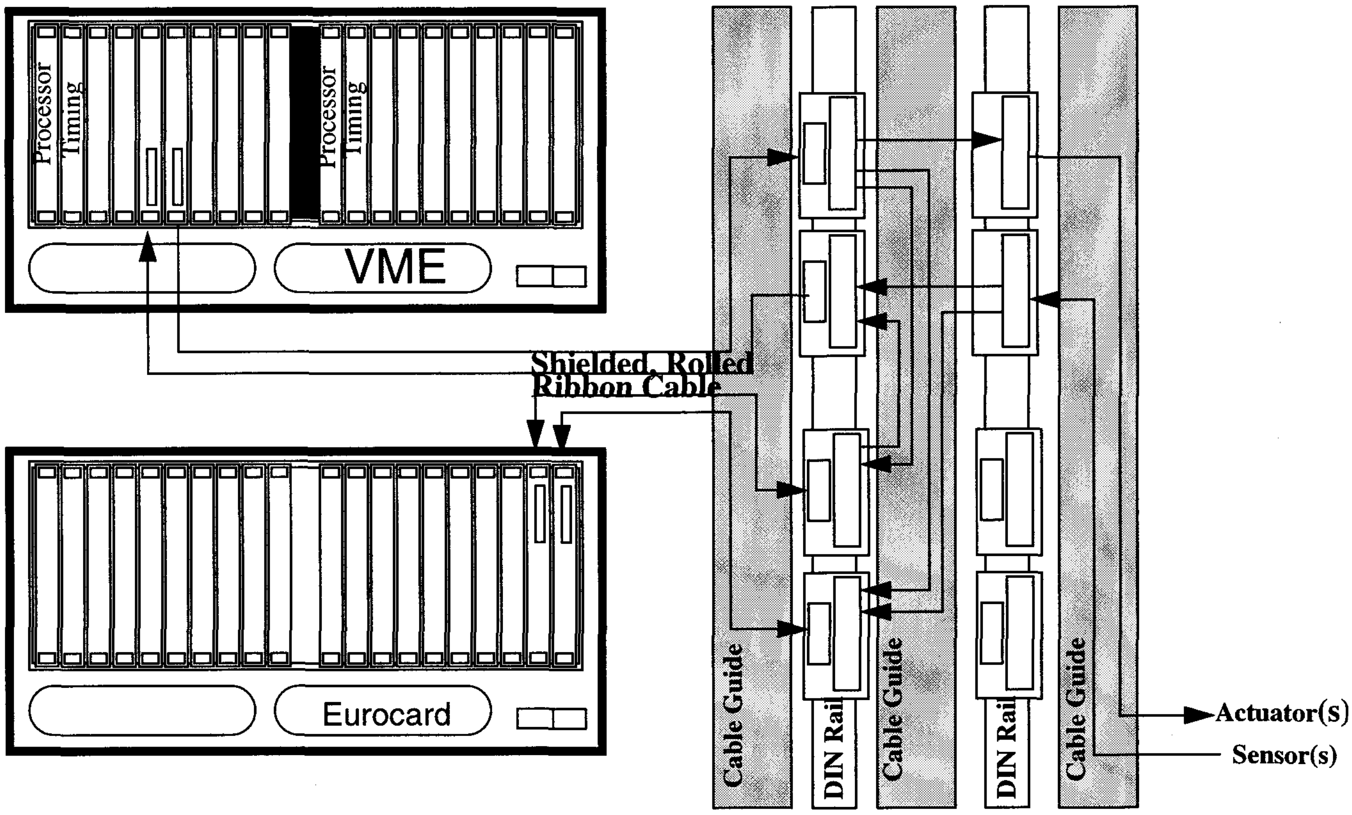



**Figure 7: CDS Cable Interconnect System**

## 3.5. Interfaces

The following figure shows the interfaces to CDS front end system racks. These are further described in the following subparagraphs.

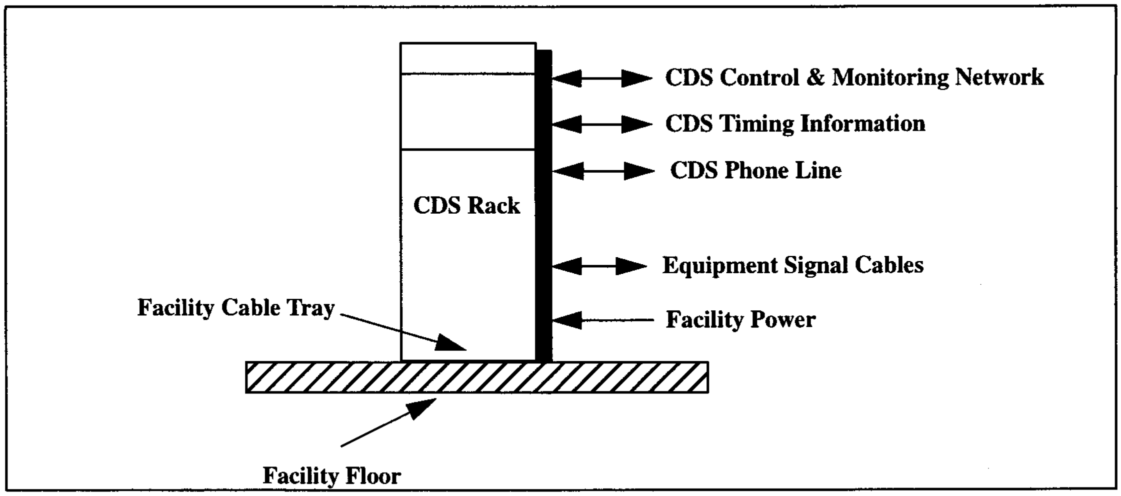



**Figure 8: CDS Front End Rack Interfaces**

### 3.5.1. Rack Placement

For the LVEA area, initial rack estimates and placements are shown in Figure 9: CDS Rack Placement within the LVEA. Mid and End station rack placements are TBD.

### 3.5.2. Cable Raceways

Cable trays will be provided (by others) under the beam tubes for the routing of CDS cables. These trays will be subdivided into three parts to provide separation of cables by function:

- Analog signal cables
- Digital cables
- Power cables

Short tray stubs will be provided at each rack location to get from the main cable trays to entry at the rack bases.

### 3.5.3. Cable in Vacuum

CDS must provide certain cables into the LIGO vacuum chambers. All cabling within vacuum will meet the Vacuum Cabling and Feedthrough (VCF) requirements as outlined in LIGO T950095-C.

### 3.5.4. Rack infrastructure cabling

Infrastructure cabling includes those cables and functions which are to be provided to all CDS front end racks. The following table shows the infrastructure cabling to be provided to each CDS front end rack pair (the normal CDS rack configuration).

**Table 1:**

| *Function* | *Cable Type* | *Qty* |
|---|---|---|
| Networking | Category 5 copper | 3 |
| ATM/FC Networking | Single Mode Fiber (pair) | 2 |
| Telephone | Category 5 copper | 1 |
| Timing | RG-58 | 1 |
| 110VAC Service | 3 conductor, 20A | 4 |

### 3.5.5. Facility Power

Two 110VAC/16A service lines will be provided to each CDS rack. In the LVEA, this service will be brought through conduits encased in the concrete flooring from 20A breaker panels on the LVEA walls at various locations. In other buildings, power cable routing is TBD.

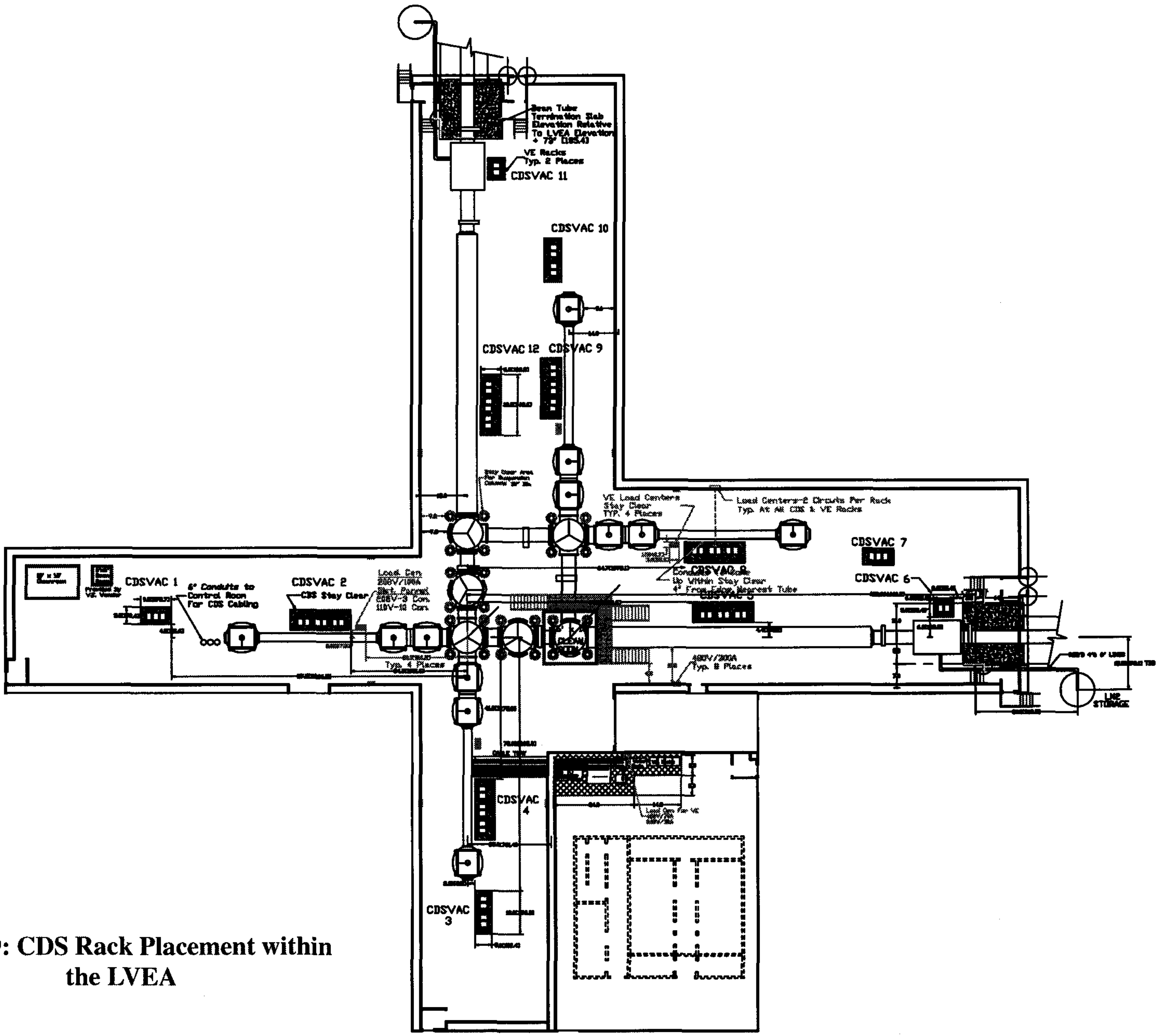



Figure 9: CDS Rack Placement within the LVEA

# 4 TIMING SYSTEM

## 4.1. Design

The CDS timing system is based on the Global Positioning System (GPS). A GPS antenna and receiver will be located at each mid and end station, along with one at the corner station area to serve the LVEA and OSB.

The system layout is shown in Figure 10: Timing System Overview. At each LIGO building area, a GPS antenna will be mounted on the roof and connected via a coax and serial communication line to a VME based Odetics TSAT-VME GPS receiver module in one of the CDS VME crates in the area. All other VME crates within the building/corner station area will house Odetics TPRO-VME GPS slave modules. The actual receiver circuitry is housed in the antenna unit. The difference between the VME receiver module and the slaves is only in that the receiver module has the added connections to accept the antenna inputs.

Once the receiver is powered up, it takes up to 30 minutes to acquire enough satellite information to get accurate time information (time dependent on how long the system has been off). Once time information is acquired, it is available:

1. To the VME processor over the backplane via direct memory locations.
2. To VME slave units via IRIG-B connections.

In addition to time-of-day information, the receiver and slave VME units produce selectable phase-locked TTL clock outputs with frequencies from 1Hz to 10MHZ. These clocks will be used as synchronization inputs to DAQ modules and control and monitoring signal digitizers.

The GPS module design also allows VME interrupts to be generated at these clock frequencies. These will be used to synchronize time critical software events.

A copy of the Introduction and Specification sections of the Odetics GPS module manual is provided in Appendix TBD for further operational information.

## 4.2. Interfaces

Interfaces to the timing system are:

1. At the VME backplane connection to the receiver/slave modules.
2. At the timing clock output jack on the module front panel.

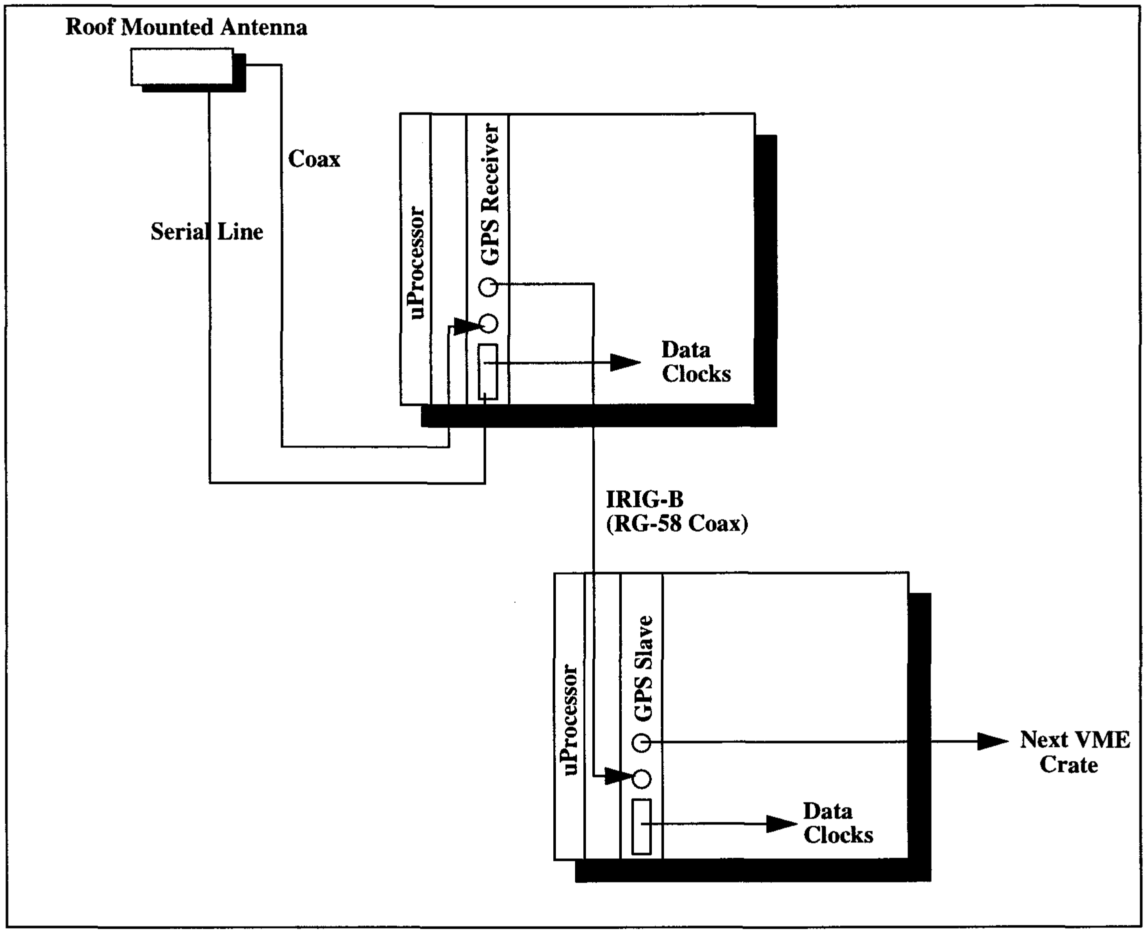



**Figure 10: Timing System Overview**

# 5 NETWORKING

## 5.1. Bandwidth Requirements Analysis

Bandwidth requirements for CDS control and monitoring are fairly minimal and could be met through the use of standard ethernet. The distances involved in LIGO would require bridging systems.

The higher bandwidth requirements come from the data acquisition system. For each interferometer, the DAQ must have the capability to acquire, transport, assemble and store data at a continuous rate of up to 6MBytes/sec. Therefore, the network must be capable of >6MBytes/sec/IFO, such that transporting the data does not take up the entire time allotment of the system. A minimum factor of 2 is desirable, resulting in a bandwidth requirement of 12MBytes/sec/IFO.

Combining control and monitoring needs and the data acquisition needs, the initial networks must therefore have a bandwidth of >200Mbits//sec to support the initial two IFO at Hanford, with expandability to at least 600Mbit/sec for six IFO.

Equally as important as total bandwidth is network throughput. Throughput here is defined as the amount of data which can be moved from the memory of one processor over a network into the memory of another processor. Both processor performance and network bandwidth contribute to this figure. Assuming that each DAQ acquisition front end processor is required to handle up to 32 channels at 20KHz and 32 channels at 2KHz, and half the time is allotted to data transfer from the front end to the frame builder, the throughput requirement from the front end processor to the frame builder is 3Mbytes/sec (24Mbits/sec).

## 5.2. CDS Network Design

Figure 11: Network Architecture Overview shows a proposed network layout using commercial networking equipment. Both Asynchronous Transfer Mode (ATM) and Fiber Channel (FC) are being studied for use as the high speed backbone of the system. Since both require point-to-point hardware connections, the architectural layout would be similar. As shown in the figure, all ATM/FC connections would connect to a central switch at the OSB. This switch allows interconnection of all ATM/FC nodes on the network. All connections to this switch are fiber optic.

Two fibers from the switch would be run to each of the mid and end stations. One fiber connects to a bridge/router, which supports ethernet connections onto the high speed backbone. This would allow connection of control and monitoring processors and the FCMS in these buildings. The second fiber will run directly to an ATM/FC VME interface module to provide direct high speed access from a data acquisition VME processor in the building.

For the LVEA, multiple fibers would extend from the switch for direct connection to data acquisition processors. Ethernet capabilities would be provided via an ethernet bridge connected to the main CDS router in the OSB.

Within the OSB, the data acquisition system frame builder(s) and server(s) would have direct connections to the ATM/FC switch. Other control and monitoring equipment, such as operator consoles, servers, etc., would be connected by ethernet and interface to the ATM/FC via the main CDS router. This router also provides access to OSB office workstations via ethernet and to other LIGO sites via T1 interfaces.

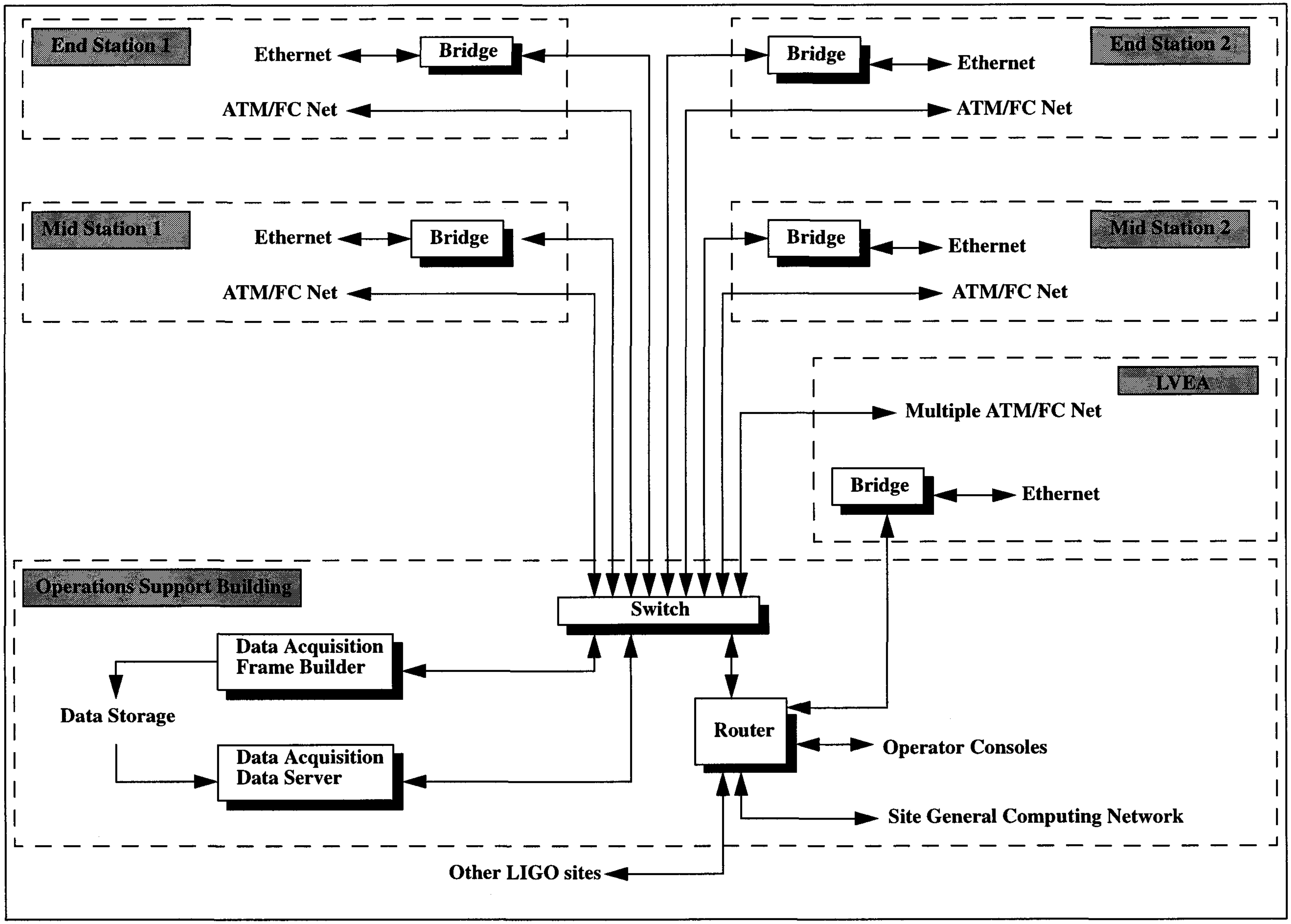



**Figure 11: Network Architecture Overview**

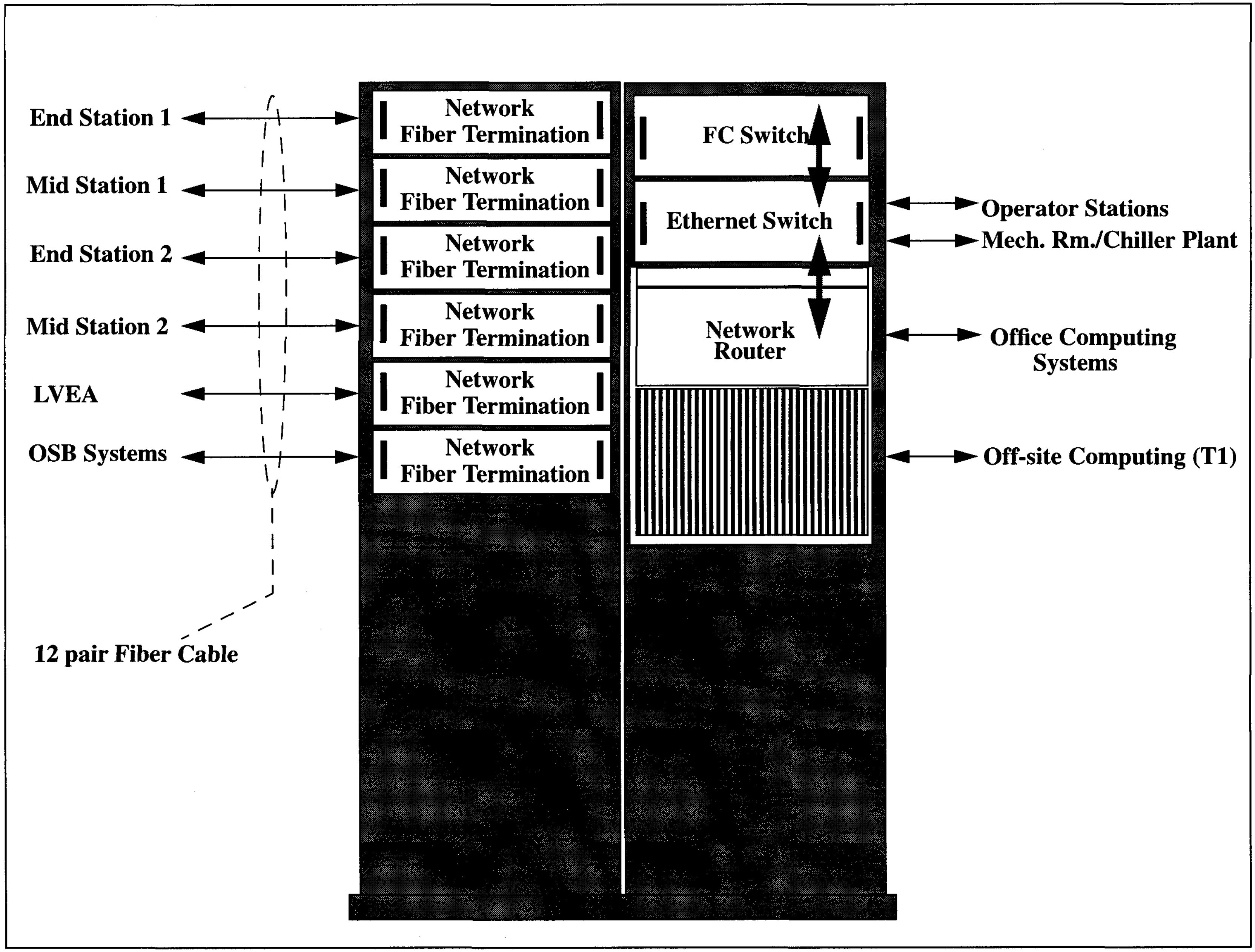



**Figure 12: Network Rack Layout (OSB Computer/Mass Storage Room)**

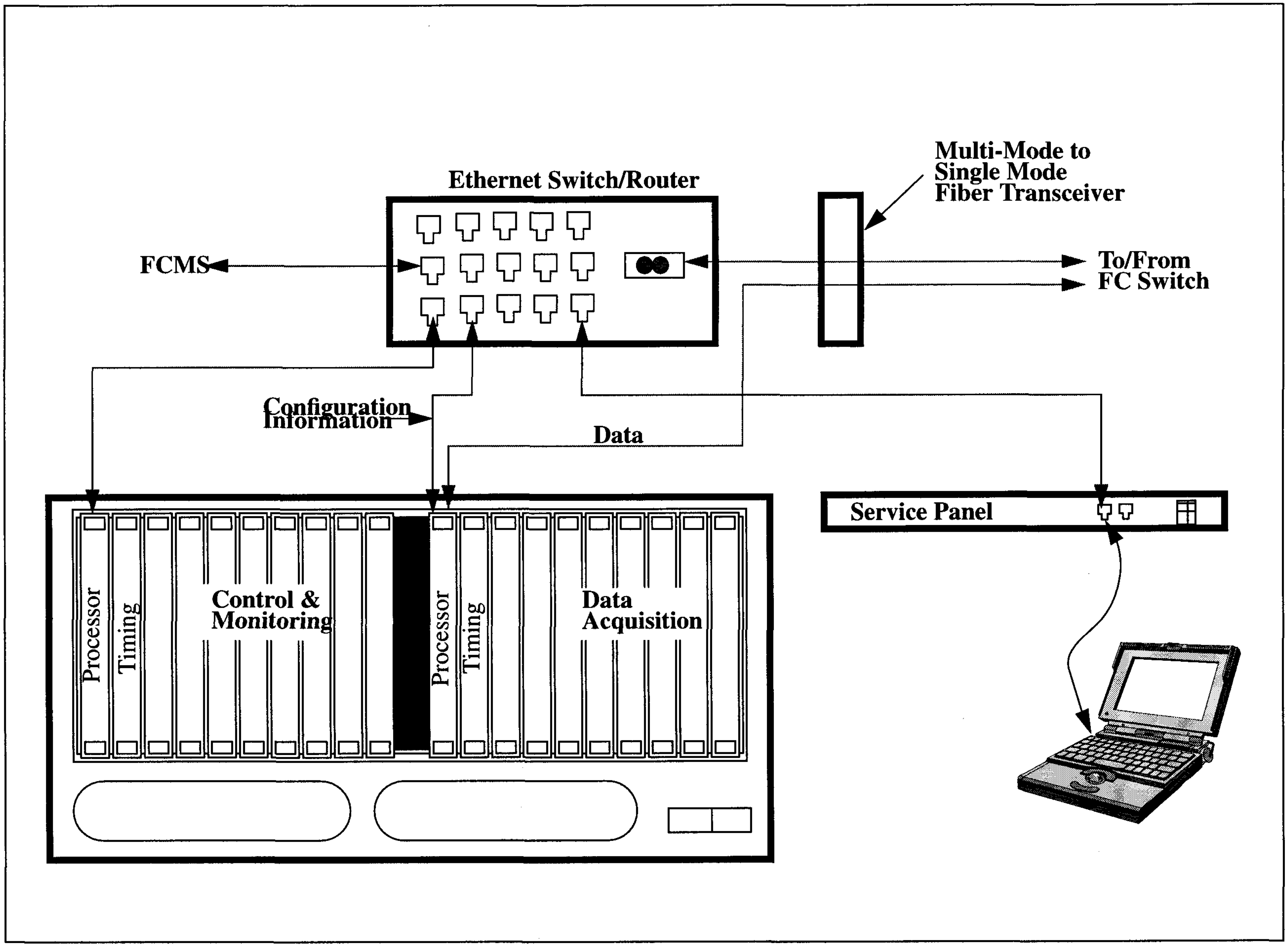



**Figure 13: Local Network Connections**

# 5.3. CDS Network Design Analysis

## 5.3.1. General

While the previous sections described a possible layout for ATM or FC, Fiber Distributed Data Interface (FDDI) networks are also being considered. The latter would require a different network layout as it is based on a token ring topology rather than a point-to-point connect system.

A final choice of network will be based on:

- Performance as necessary to meet LIGO requirements, in terms of both overall bandwidth and throughput between processors.
- Availability of software drivers.
- Commercial hardware support.

The following table summarizes the present status of these network types in respect to performance.

**Table 2: Network Media Comparisons**

| *Type* | *Supported Data Rate* | *Node-to-node Throughput (Average)* |
|---|---|---|
| FDDI | 100 Mbits/sec | 30 Mbits/sec<br>(Between two HP Risc Computers) |
| ATM | 155 Mbits/sec<br>(620 Mbits/sec next year) | 48 Mbits/sec<br>(Between two HP Risc Computers) |
| FC | 1062 Mbits/sec | 160 Mbits/sec (Between two Sun super sparcs)<br>400 Mbits/sec (Between two Sun ultra sparcs) |

## 5.3.2. ATM and FC Overview

Both ATM and FC provide data transport systems which are partly network orientated and partly connection orientated. A connection based system requires a network manager to establish point to point links between all communicating nodes before data transfer can begin. Network systems, on the other hand, can dynamically establish links between nodes if and when the need arises. Networks use a variety of protocols to accomplish this, including Address Resolution Protocols (ARP) and a variety of routing protocols (RIP, DISCOVERY, etc.).

In some respects, LIGO's data transfer requirements can also be thought of as partly channel, partly network based. The Data Acquisition System requires dedicated, high bandwidth data channels to be established between known data producers and data consumers. The Control and Monitoring EPICS system, on the other hand, demands the full networking infrastructure supporting TCP/IP streams and UDP/IP broadcasting.

### 5.3.3. ATM

ATM stands for Asynchronous Transfer Mode, and arose in the telecommunications industry from the older Synchronous Transfer Mode (STM) standard.

STM uses time division multiplexing to send several channels of data over a link with guaranteed bandwidth and latency. Unfortunately, for computer data (which is usually bursty in nature) this means that for most of the time the link is being used very inefficiently. Therefore the ATM standard was introduced to share the available bandwidth amongst competing data producers more fairly.

ATM is a connection orientated system. This introduces some problems when it is used to provide a network service, as will become apparent.

Connections between ATM nodes are either Permanent Virtual Connections (PVC) setup by network management prior to data exchange, or are Switched Virtual Connections (SVC), dynamically established when data transfer is required. PVC's and SVC's are channel services and network services respectfully. Only SVC's will be discussed in this document.

SVC's are set up using the "one pass" mode in which an end node issues a request for a connection to another node. Within this request are details about the end node addressing, required bandwidth and latency, and the required Quality Of Service (QoS). This connection is issued through the node's ATM port, called a UNI. It is typically sent to an ATM switch via its NNI port. The switch forwards the request, node by node, until the end node is reached. At this stage the end node may accept or refuse the connection. If accepted, an acknowledgment is sent back along the same data path signaling that data transfer can begin.

Connections can be either point-to-point, in which case full duplex transfer can proceed, or point-to-multipoint, in which case only uni-directional transfer is allowed. This limitation is caused by the adoptance of the ATM Adaptive Layer AAL5 standard, and is the reason why multicasting cannot be directly implemented to provide network services over ATM.

To provide multicasting, several solutions (or fudges if you will) are available. The two most popular are Multicast Server and Overlaid Point-to-point.

In the Multicast Server option, an external multicast server is introduced onto the network. All nodes establish point-to-point connections with the server, which then establishes point-to-multipoint connections with all other nodes. Thus the server is a go-between offering multicasting services to all nodes.

The Overlaid Point-to-multipoint solution requires that all nodes establish point-to-multipoint connections to every other node on the network.

The multicast server option is more commonly adopted since it cuts down on the number of channels which have to be supported. Both solutions require knowledge of which nodes are on the net, and network management intervention is required to establish the connections. This may cause problems if the number of nodes on the net is not constant.

ATM has adopted an overlay model of addressing, essentially creating its own network layer below the Internet Protocol ISO layer 3. The ATM sub-layer address would appear to be a Network Service Access Point (NSAP), but should really be thought of as an ATM End-point-identifier.

| ISO representation | | |
|---|---|---|
| | IP | level 3 |
| | ATM NSAP | level 2 |
| | Physical | level 1 |

LAN services are supported on ATM networks through the use of a LAN Emulation (LANE). This provides a complete emulation of an existing LAN (ethernet or IEEE token ring) between the ATM "network" and the IP layer.

| ISO representation | | |
|---|---|---|
| | IP | level 3 |
| | MAC | level 2 |
| | ATM NSAP | level 2a |
| | Physical | level 1 |

The introduction of the new Media Access (MAC) layer does not fit within the standard ISO representation, but this is hardly surprising since we effectively have a "network in a network".

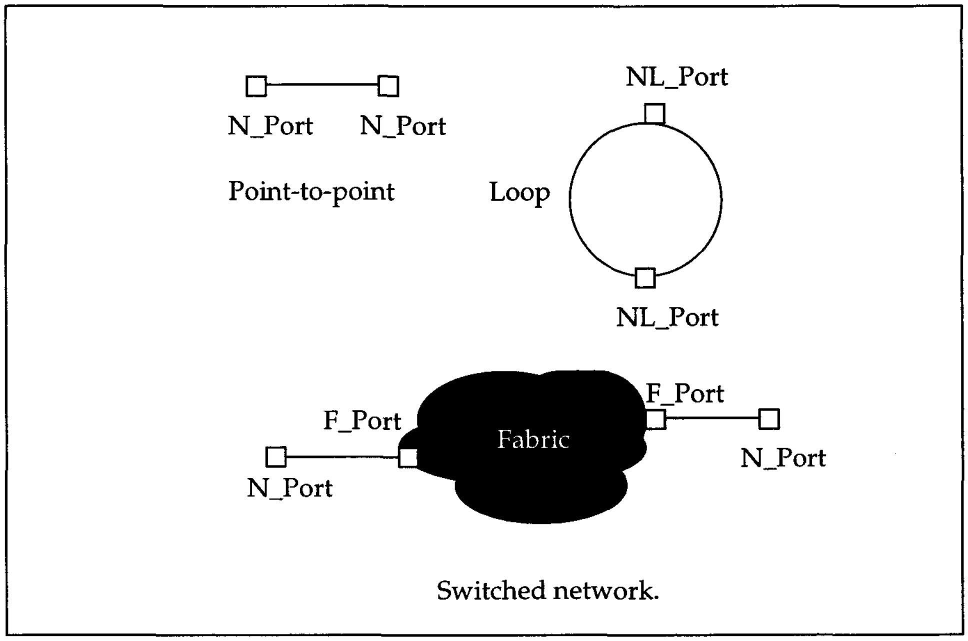
ATM does not provide guaranteed delivery of data end-to-end, and relies on higher level protocols (e.g. TCP) to provide these services. This is a direct result of ATM's telecommunications origins, where the requirements allow for some data loss, but does not tolerate data delay. This conflicts with computer digital data transfer which tolerates data delay, but not data loss.

Current ATM bandwidths are at the OC3 stage of 155Mbps per channel. There are promises for future extensions into the 1Gbps range.

### 5.3.4. Fiber Channel

Fiber Channel (FC), like ATM, provides a range of services which are a mixture of channel and network based services.

FC provides a high bandwidth per channel (1Gbps up to 10km on single mode fiber optics). FC is topology independent, running over point-to-point, loop and switched networks. End node FC ports are called N_Ports. If they are in a loop topology they are called NL_Ports. A switching system is referred to as a fabric, and its ports are called F_Ports.




There are several classes of service;

- Class 1 is connection oriented. After connection between hosts has been established, full bandwidth (100MBps full duplex) is available to the channel.
- Class 2 does not require a connection to be established before data is transmitted. Data is sent frame by frame.

Both class 1 and class 2 services provide the sender with full frame reception acknowledgments. In addition, class 1 guarantees that frames will be delivered in order. Both classes provide a guaranteed delivery service.

- Class 3 is connectionless, and no confirmation is given of data reception. Data delivery is not guaranteed with this service.
- Several data users can send data on a single channel using the hybrid class called intermix. This uses a class 1 service to provide guaranteed bandwidth for a privileged user, but makes any unused bandwidth available to the other users with classes 2 and 3 services.

The FC protocol defines 5 layers which fit below the ISO network layer;

| | | |
|---|---|---|
| ISO representation | IP | ISO level 3 |
| | FC4 | NSAP |
| | FC3 | Multicasting |
| | FC2 | Framing |
| | FC1 | En/Decoding |
| | FC0 | Physical |

FC layer FC4 provides a Network Service Access Point between FC and a variety of higher level protocols. These include SCSI, IPI, HIPPI, IP, AAL5 (ATM) and IEEE 802.2.

FC layer FC3 provides some network services, including FC multicasting.

FC layer FC2 performs the framing of the data into FC frames.

FC layer FC1 performs the 8B/10B encoding and decoding. Each 8 bit block of data is encoded into 10 bits. This adds clock synchronization data. The quoted bandwidth of 10MBps includes this 25% network overhead.

FC layer FC0 is the physical layer. It provides the drivers for the fiber optics lasers. It is interesting to note that the laser power to drive 10km of cable is such that safety measures have been included at this level. If a break in the fiber line is detected, output power is dropped to safe levels.

Fiber Channel provides IP applications with a full suite of network features, including address resolution, broadcasting and the infrastructure for streams (connection) and datagram (connectionless) services.

### 5.3.5. Differences between Fiber Channel and ATM.

If both ATM and FC adhere to the networking standards for IEEE 802.3 (Ethernet) and TCP/IP in their respective LAN services, then both networks will appear to be identical as far as the application is concerned.

They differ in the amount of network management which is needed to set up the network, and the number of hard ware units needed. In this respect, it would appear that the ATM solution is both more hardware intensive and software intensive. If the multicast server solution is selected (as recommended by Cisco) then an extra router is required to handle this task. It is not clear how the router will manage nodes joining and leaving the network.

The FC design lends itself more directly to networking. The main features for networking are;

- Multicasting capability for broadcasting.
- Connection orientated links for TCP/IP streams.
- Connectionless links for UDP datagrams.

FC provides all of these services.

Regarding VME cards for ATM and FC. CEBAF is currently writing VxWorks drivers for their Interphase ATM card in collaboration with NASA Goddard. This effort is geared towards using ATM for CODA Data Acquisition data transfer.

There are no known VxWorks driver for the Anchor FC VME card.

ATM and FC networks appear to be industry frontrunners for high performance networks in the near future. ATM systems presently operate at OC-3 rates (155Mbits/sec), with OC-12 (620Mbits/sec) expected in the next few years. FC systems run at up to 1.2Gbits/sec, with 255Mbits/sec the present offering for a network fabric.

ATM is based on point-to-point connection service developed by the telecommunications industry. Due to its high bandwidth and long distance capabilities, numerous data communications vendors have been working on developing protocols to allow use of ATM for data networks. This has involved a vast amount of software development, as standard networks are typically connection-less, packet switched and ATM requires a point-to-point connection and is time domain multiplexed. At present, several vendors do offer the necessary switching and interfacing equipment, along with "virtual" LAN software to provide standard network protocol connections. The virtual LAN software runs in an ATM-equipped network router to provide what appears to the end user as connection-less, packet switched access and broadcast capabilities. Some key issues in the use of ATM are:

- High cost - presently double that of a similar FC system
- Virtual LAN - While running on ATM, does not provide for ATM Quality of Service (QoS), which allows priorities to be set for transmissions. This would be useful in establishing deterministic data channels for real-time control across networks and allotting bandwidth to high priority tasks.
- Standards - ATM standards are still emerging for use for data communications. Present offerings are all proprietary and do not allow intermixing switch and router equipment from different companies. Many vendors are involved in the various standards committees, which sometimes slows the process.

FC is also a point-to-point network, but not based on a telecommunications industry standard. Rather, it was defined and standardized to provide more data network features, which include:

- Direct Connect, which creates a point-to-point channel, allowing large data buffer transfers directly between two processors on the net.
- Connection-less service, allowing for standard networking packet-switching interfacing.
- Broadcast service, allowing for standard network broadcasting of messages throughout the FC fabric.
- Intermix service, allowing both direct connect and connection-less service simultaneously to a FC node.

The key issue in the use of FC is the limited number of vendors which provide the FC networking equipment. Only a few offer FC switches. Most FC work by various companies appears to be in the area of replacing HiPPI interfaces with FC interfaces to their high speed data storage systems.

### 5.3.6. Initial Approach

The initial approach will be to develop a network based on FC. FC presently has a cost and performance advantage over ATM. Emergence of both FC and ATM commercial equipment will continue to be monitored before a final selection is made for the LIGO sites.

FC routers exist with the necessary software to interconnect standard networks, such as ethernet, fast ethernet, FDDI, etc. For the CDS VME implementation, hardware exists, but drivers would need to be developed to make the network connections directly to the VxWorks operating system and into the CDS control and monitoring software. However, since normal control and monitoring only requires ethernet bandwidth, these processors will connect to FC through standard routers for the present. Since TCP/IP is supported for FC on hardware platforms for which software drivers exist, it is likely, over time, that TCP/IP driver software will be developed commercially for VxWorks or similar real-time operating systems. As long as the TCP/IP standards are maintained, it should be a straight hardware replacement at the time of implementation, if direct connect to FC is required and/or desired for high performance control needs.

FC modules are now starting to be produced for PCI bus. CDS real-time front end processors will have PCI mezzanine (PCM) slots to accommodate these modules where necessary.

For CDS data acquisition, where the high bandwidth of FC is required, Sun Sparc VME processors will be used with S-bus FC modules. Software drivers do exist for these units under the Solaris operating system. The real-time extensions and multi-threading capabilities of Solaris appear to make this a viable solution.

## 5.4. Rack Layouts

Figure 6: Standard Rack Layout (Mid and End Stations) depicts the Networking and Fiber Termination equipment within the two bay rack systems. Figure 12: Network Rack Layout (OSB Computer/Mass Storage Room) shows how the networking equipment for the OSB would be installed. For the LVEA, a rack located near the entry point of the 6" conduits from the FCR would be used to distribute network and telephone lines, as shown in Figure 13: Local Network Connections.

## 5.5. Networking Cable Plant

The following table describes the network and telephone cable plant for a LIGO site.

**Table 3: CDS Network Cable Plant**

| *From* | *To* | *Function* | *Type* | *Qty* |
|---|---|---|---|---|
| FCR | LVEA Net Rack | ATM/FC/Ethernet | Fiber Optic (12 pair) | 1 |
| FCR | Mech. Rm | Ethernet | Category 5 Copper | 4 |
| FCR | Chiller Plant | Ethernet | Category 5 Copper | 2 |
| FCR | Mech. Rm | Phone Lines | Category 5 Copper | 2 |
| FCR | LVEA Net Rack | Phone Lines | Category 5 Copper | 12 |
| FCR | Mid Station 1 | ATM/FC Network | Fiber Optic (12 pair) | 1 |
| Mid Station 1 | End Station 1 | ATM/FC Network | Fiber Optic (12 pair) | 1 |
| FCR | Mid Station 2 | ATM/FC Network | Fiber Optic (12 pair) | 1 |
| Mid Station 2 | End Station 2 | ATM/FC Network | Fiber Optic (12 pair) | 1 |

# 6 OPERATIONS SUPPORT

An overview sketch of the CDS areas of the OSB are shown in Figure 14: OSB Conceptual Design (Partial). The CDS portions are shown shaded. These areas include:

- A Facility Control Room (FCR), from which normal LIGO operations are carried out.
- Computer/Mass Storage Area, which contains compute servers and disk drive and tape units.
- An Electronics Shop, where CDS systems are developed and maintained.

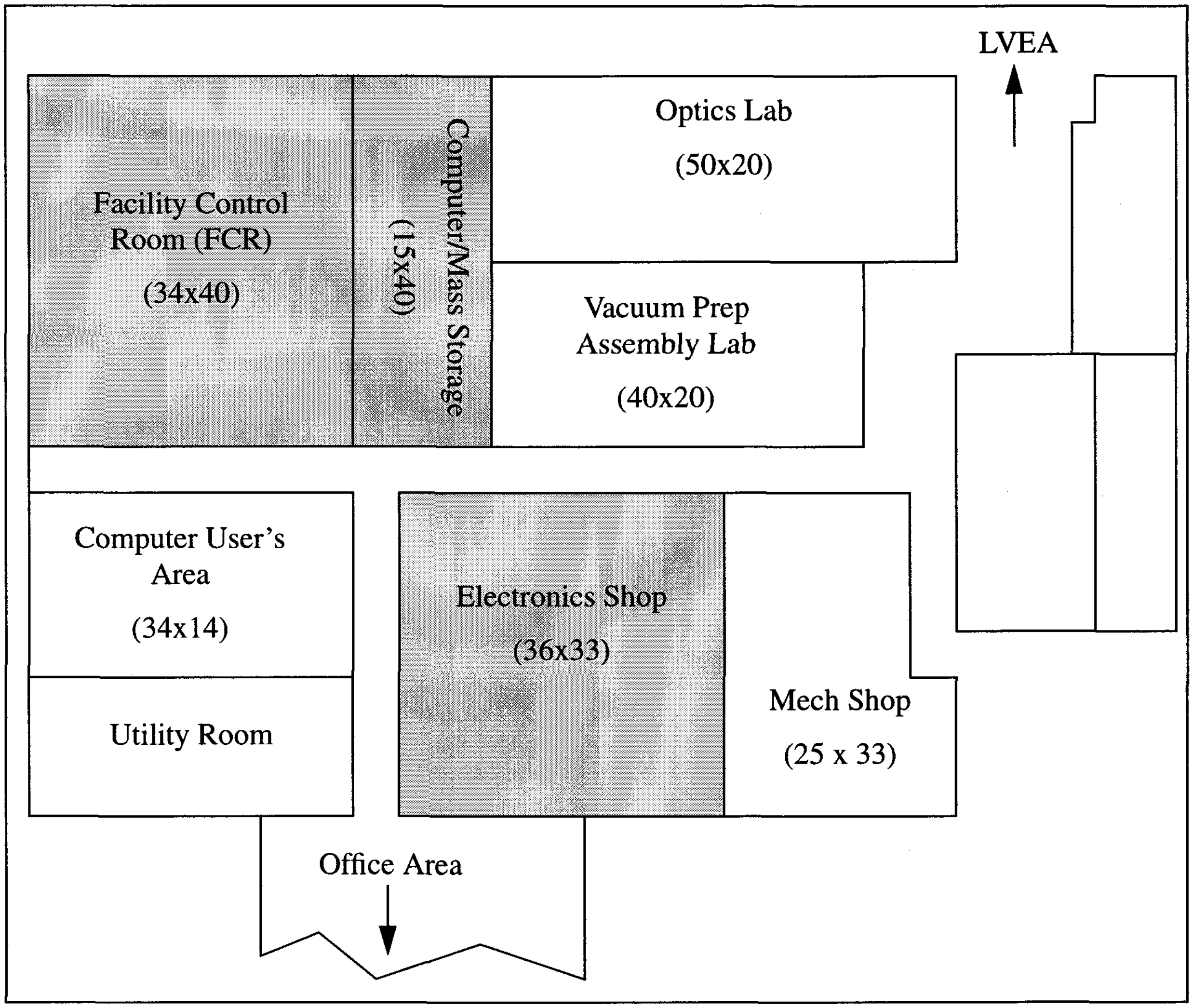



**Figure 14: OSB Conceptual Design (Partial)**

## 6.1. Facility Control Room

A concept for the FCR is shown in Figure 16: Facility Control Room (Plan View) and Figure 17: Facility Control Room (End View). The FCR features:

- A "video wall", consisting of 3 each 84" diagonal monitors. These are high resolution through-the-wall projection systems, equipped with interfaces to the various control room monitors and a switching system allowing views to be presented from any of the operator workstations. These monitors would typically display overview status of LIGO systems, but may be selected from operator consoles to show any of the LIGO CDS display pages, along with video.

- Five operator consoles/stations. During the LIGO commissioning phase, the five consoles provide space for the larger engineering and scientific staff typically involved in commissioning activities. As LIGO moves into more steady-state operations, the two stations furthest from the video wall are intended as the main consoles for the two LIGO operators on shift, with the other three remaining consoles available for additional staff performing machine studies and tuning activities.
- Standard office furniture, such as conference/layout tables and bookshelves.

## 6.2. Operator Stations

### 6.2.1. FCR Consoles

The operator consoles consist of: low bay 19" rack units (24"W x 42"H x 30D"). These units have lighting in overhang space to provide lighting to the counter top area. Interspersed with the rack units will be desktop units of the same design, some with glass tops with recessed areas for printers.

Each console will have up to four bays for mounting of high resolution monitors, telephones, test equipment, and book storage. Both Unix and PC workstations would be employed to provide computing for the operator stations. Primary operator interface to the interferometer systems will be via the UNIX workstations. These units will provide the interactive GUIs, along with video in X windows. The PC workstations provide access to the FCMS, as well as control and data acquisition databases.

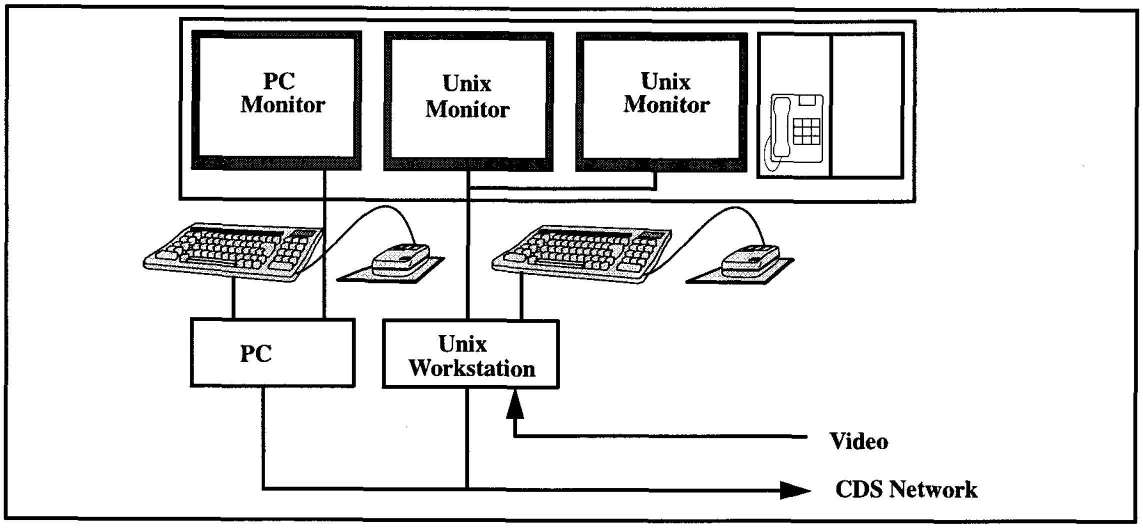



**Figure 15: Operator Console Computer Equipment**

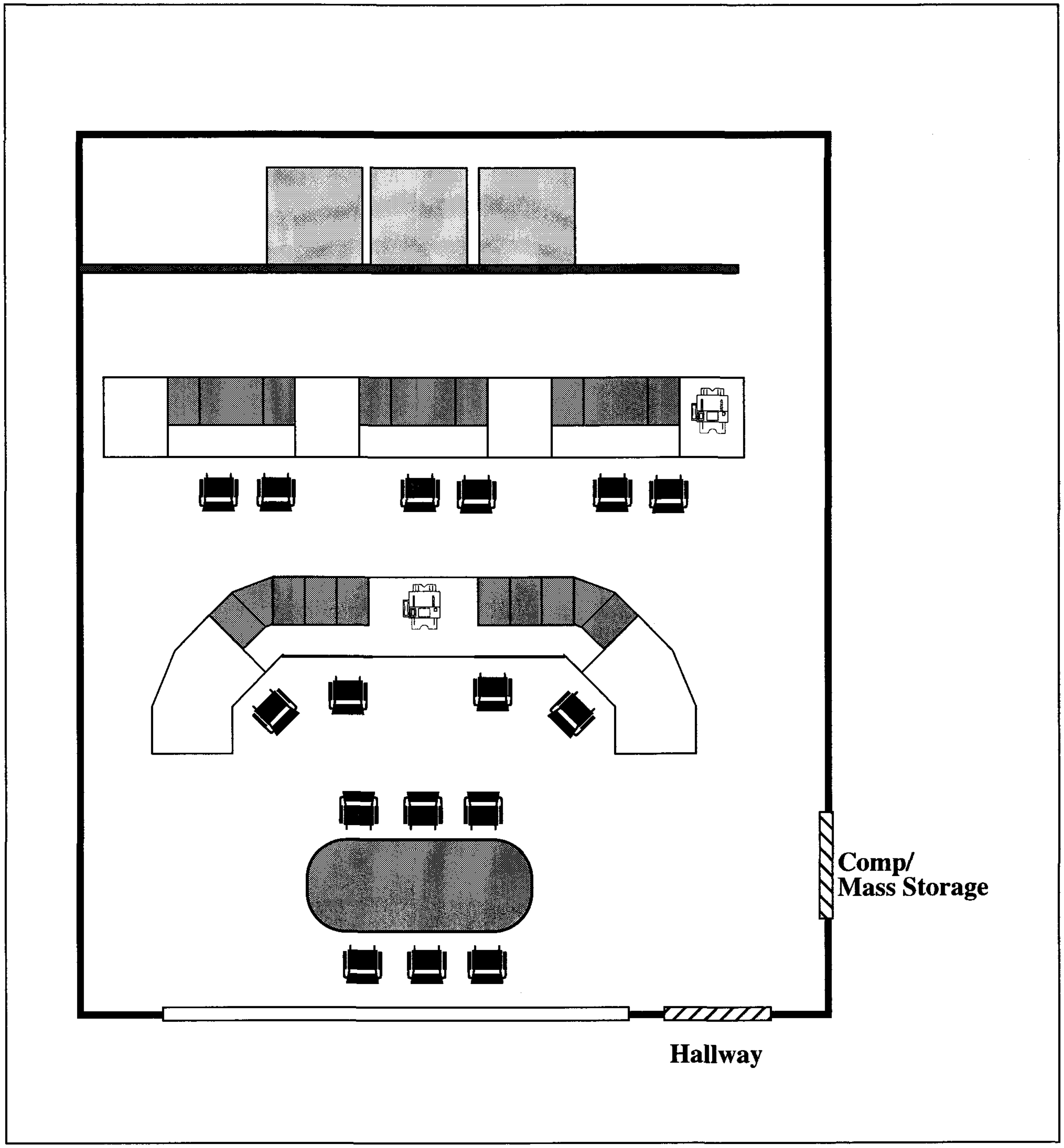



**Figure 16: Facility Control Room (Plan View)**

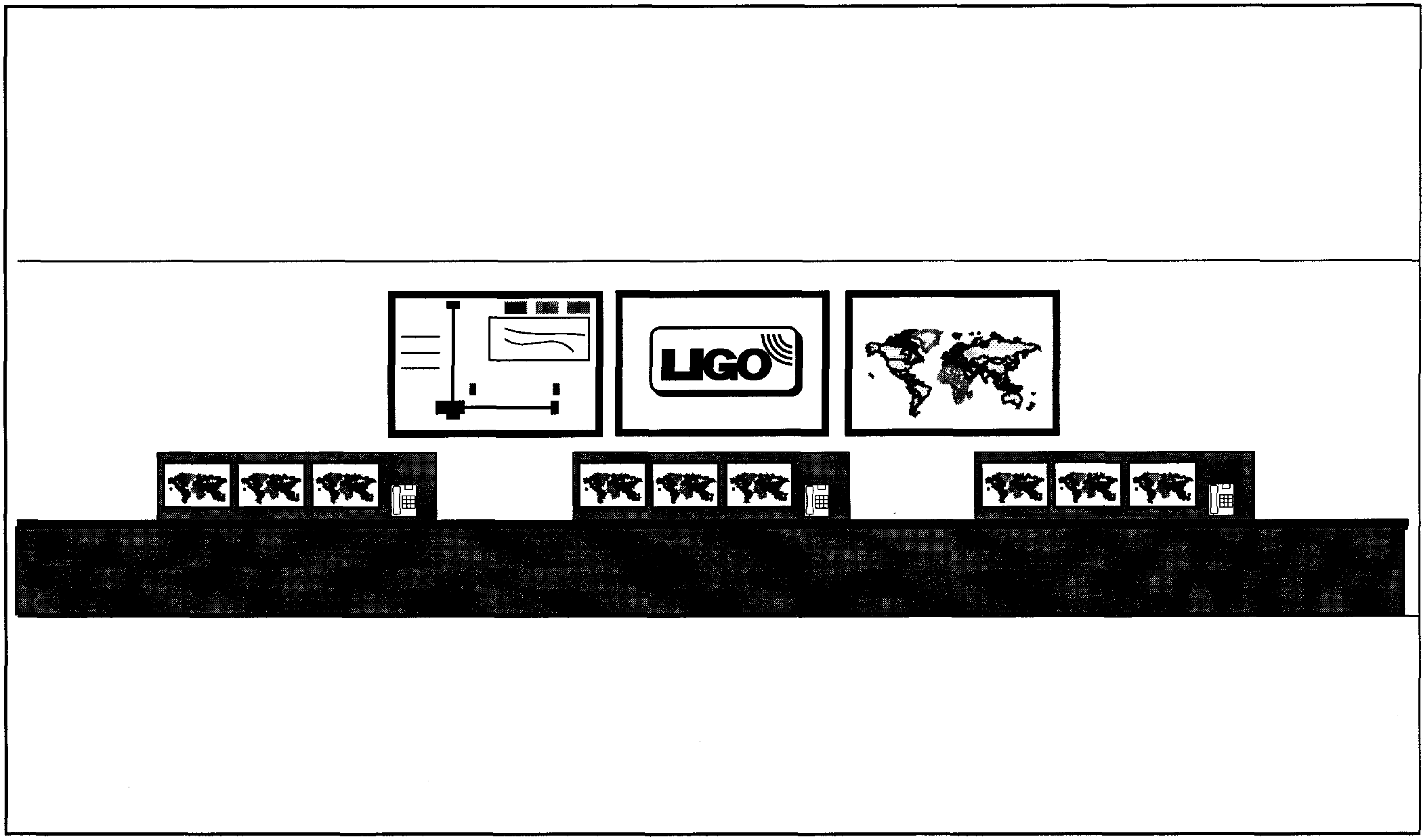



**Figure 17: Facility Control Room (End View)**

### 6.2.2. Local Operator Stations

For operation local to the equipment being controlled/monitored, the CDS racks provide an ethernet connection for use by lap top personal computers. These computers will be provided with networking and X window software to allow viewing of any CDS displays available on the network.

### 6.2.3. Remote Access

Via the CDS main network router, LIGO staff within OSB offices and from other LIGO sites will be provided access to CDS displays and information. This "outside" access will be provided at a lower priority to ensure the CDS system does not become overburdened by external systems.

## 6.3. Computer/Mass Storage Area

The Computer/Mass Storage (CMS) area will house the control and monitoring and data acquisition compute servers, mass storage units, networking equipment, and other CDS support equipment. Figure 18: Computer/Mass Storage Area shows a conceptual layout and rack assignment.

### 6.3.1. Control and Monitoring Server

The control and monitoring server will consist of:

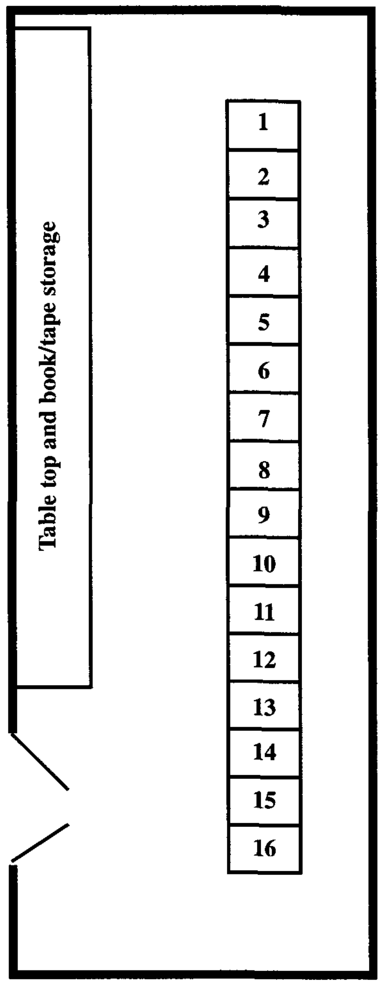
- One Sparc1000 (or equivalent) computer
- One 30 GByte RAID disc unit
- Two backup tape units
- One CD units

### 6.3.2. Uninterruptable Power Supplies (UPS)

UPS will be provided as necessary to keep the following operational during a power outage for up to 30 minutes:

- Control and Monitoring and Data Acquisition System servers
- Network hubs
- One operator console and its associated equipment

For real-time, front end processors, their criticality will be analyzed as part of the subsystem design, and UPS provided if deemed necessary and appropriate.




1. Networking
2. Networking
3. Test Equipment
4. UPS equipment
5. Disc Drives
6. Disc Drives
7. Control & Monitoring Servers
8. Data Acquisition Servers
9. Data Acquisition Servers
10. Tape Drives
11. Tape Drives
12.
13.
14.
15.
16.

**Figure 18: Computer/Mass Storage Area**

# 7 SOFTWARE

## 7.1. Overview

### 7.1.1. Real-time and Non-real-time

For control and monitoring, both real-time and non-real-time software will be developed. Software will be described in the following sections by these two categories. Real-time software is defined as that software which is deterministic in its task scheduling and duration. All software in CDS front end processors and other processors where timing is critical will employ a real-time operating system and code.

Non-real-time software is non-deterministic in its scheduling and task duration. This will typically be employed in operator interface systems and for similar functions which do not require precision timing.

### 7.1.2. System Software

System software is defined as that software infrastructure commonly used to build and run all CDS applications. This includes the software components shown in Figure 19: CDS Software Components. This would be all the operating systems and kernels on top of which all CDS software applications run, tools for use in the development of CDS software applications, and a common set of applications for typical CDS tasks, such as Human-Machine Interface (HMI), data archival, alarm management, etc.

The philosophy employed to define how this system software is produced includes the following:

1. First, and foremost, meet the LIGO requirements.
2. Use third party development tools, to the extent possible, such that code development can be more productive, particularly given the small CDS group size budgeted. These tools should provide:
   - Maximum productivity
   - Outside support i.e. a commercial product or "free-ware" which is supported by third parties. With many products available commercially and again the limited CDS group size, it is not desirable to build new tools.
3. Standardize as much code as possible and structure it for reuse.
4. Provide for cross-platform support. Being an extended project, new computers and technologies will come about over the life of the project. The concept is to develop/use software which will be a portable as possible.

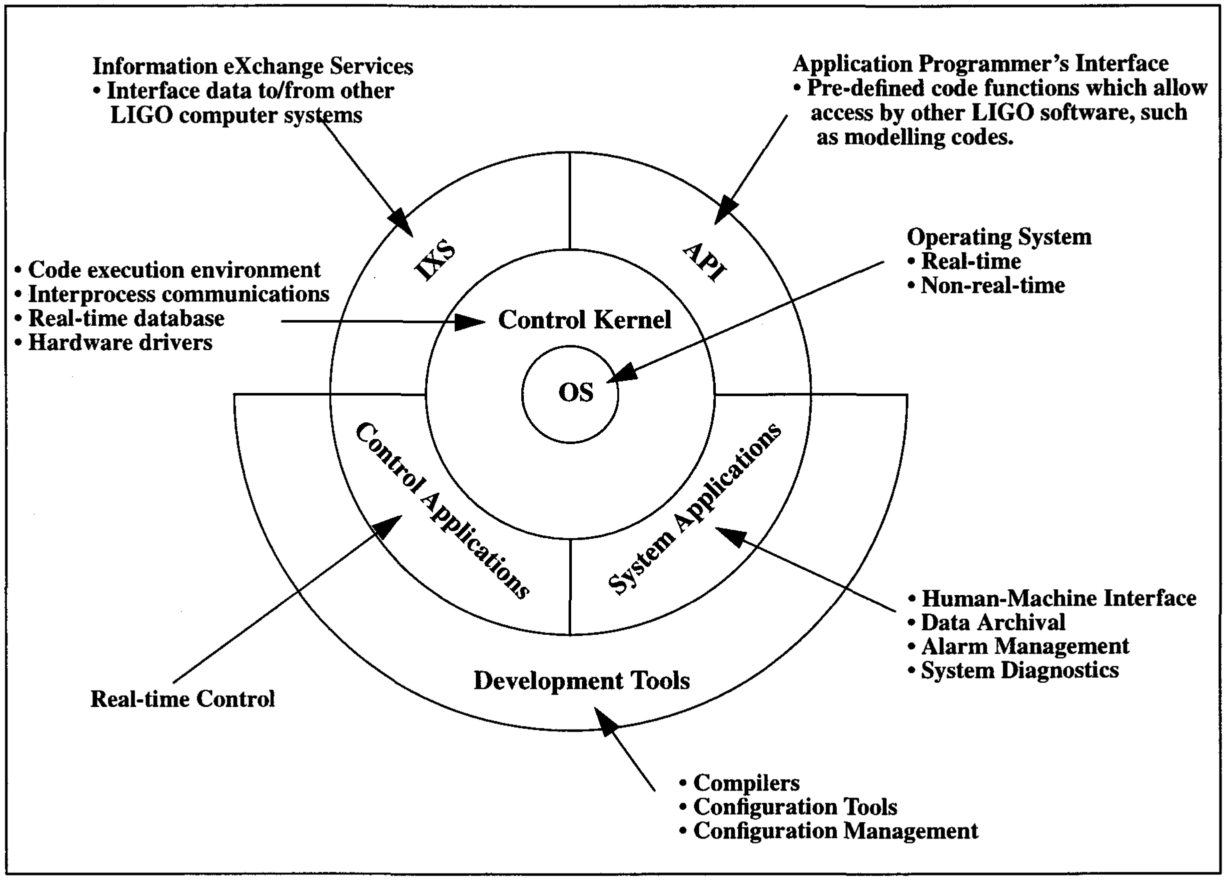



**Figure 19: CDS Software Components**

### 7.1.3. Application Software Development

Application software will be designed using a CIM model. This is the same structure which will be employed in all CDS development, including electronic systems design. A sample is shown in Figure 20: PSL CIM Model Example.

Specific application software designs are not included in this document. Requirements for CDS high level applications and resulting designs will be covered in separate documents.

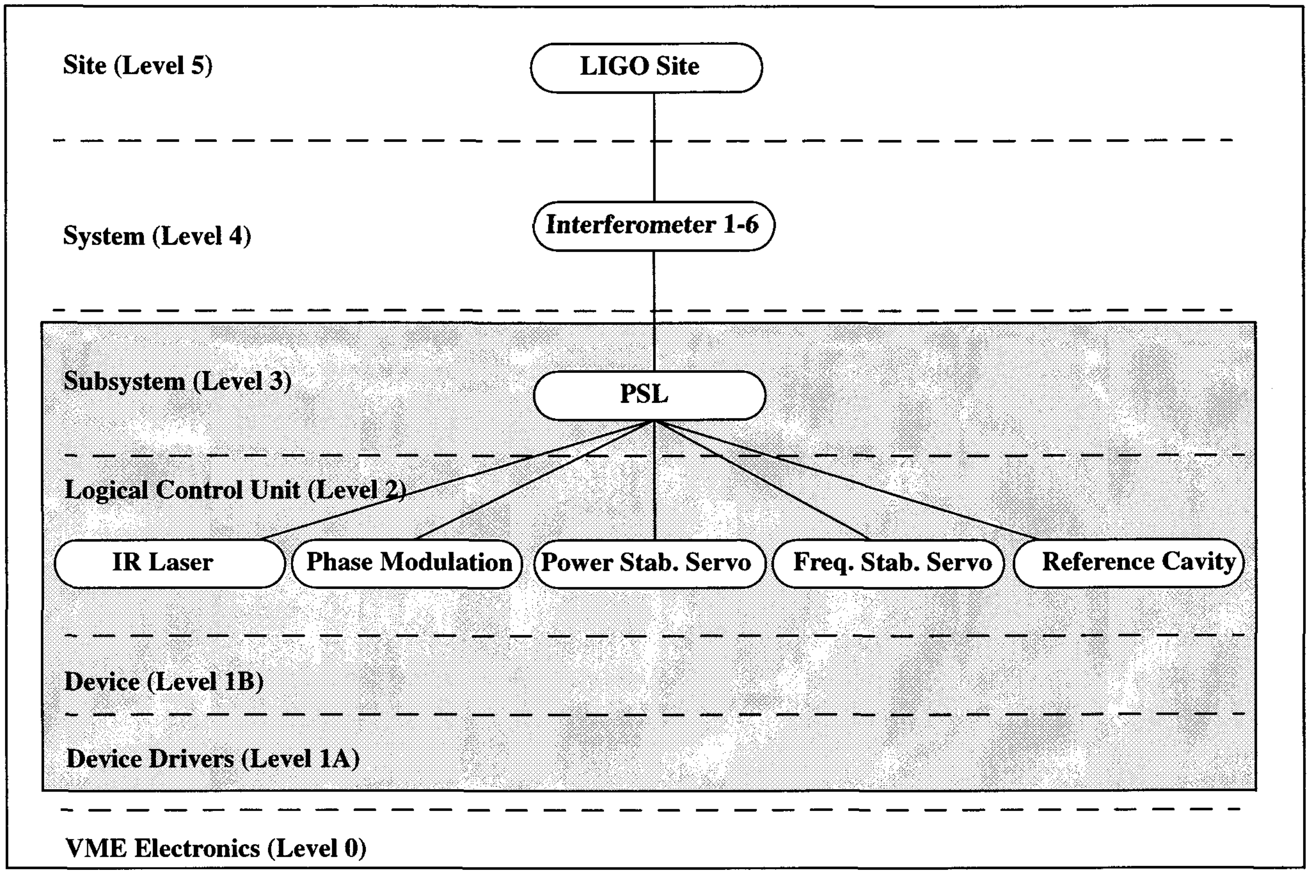



**Figure 20: PSL CIM Model Example**

## 7.2. Real-time Software

### 7.2.1. Overview

Figure 21: Real-time Software Overview depicts the primary components for the CDS real-time software development and execution. This environment will be used for all processor systems performing real-time control, which includes on processors in front end systems (described previously).

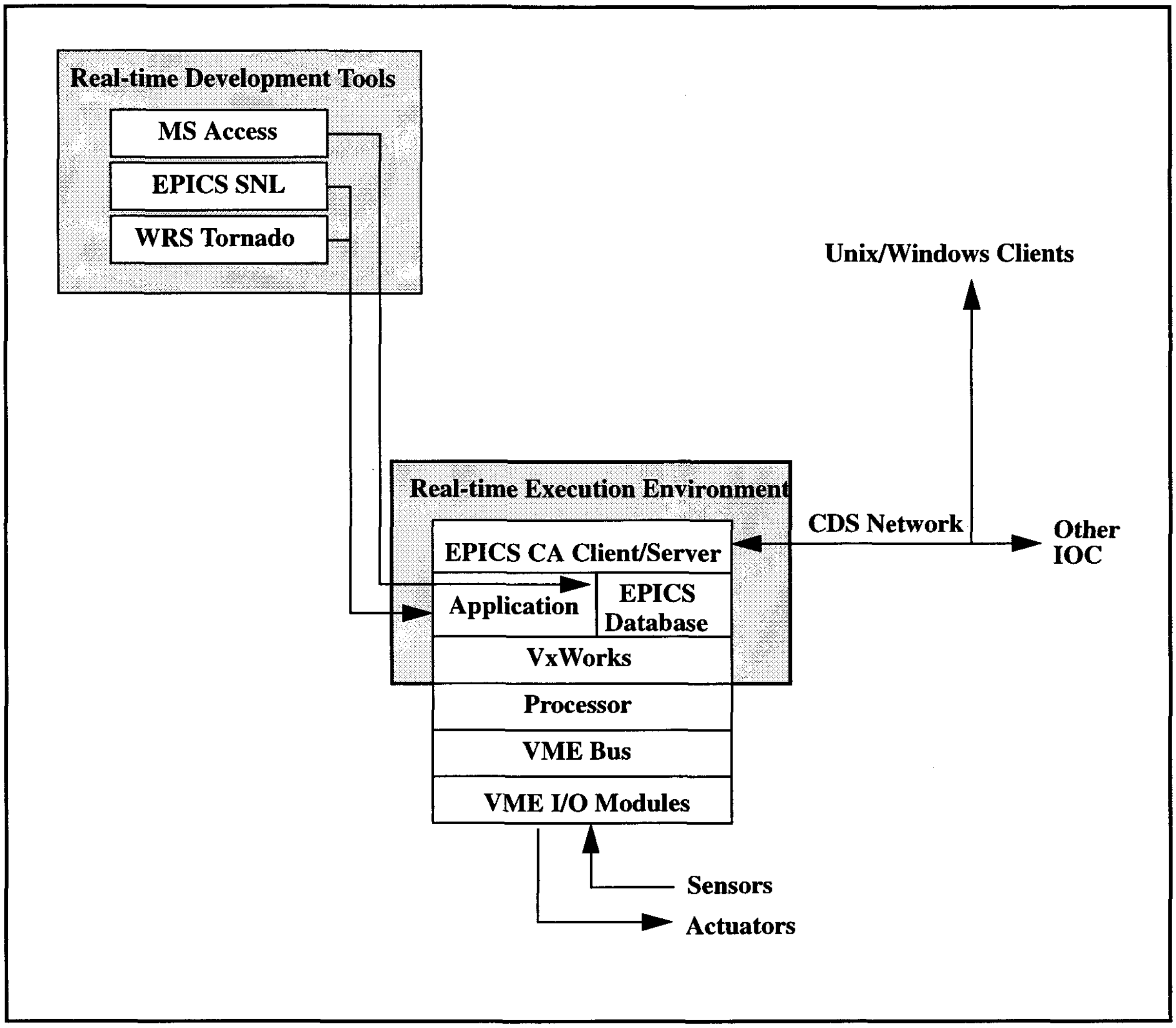



**Figure 21: Real-time Software Overview**

## 7.2.2. Operating System

### 7.2.2.1 Design

The real-time operating system to be used is VxWorks from Wind River Systems. Cross-compilers and development tools from the latest version (presently the Tornado product (VxWorks 5.3) will be used to develop and compile CDS real-time applications.

### 7.2.2.2 Design Analysis

VxWorks has a high market share in real-time software development, with very good development tools and support. The choice of VxWorks was initially driven by a decision to use the Experimental Physics and Industrial Control System (EPICS) (described later), which runs on VxWorks, but the choice of a real-time OS would have been the same independent of that fact.

## 7.2.3. Controls Kernel

### 7.2.3.1 Design

The control kernel on which real-time applications operate and/or communicate through is EPICS. EPICS provides:

1. A real-time database mechanism and associated development tools.
2. Drivers to interface a number of VME, VXI and PLC I/O modules.
3. Networking interfaces through its Channel Access modules, which allows interconnection between various real-time processors and Unix workstations via ethernet or any media providing TCP/IP.
4. A State Notation Language (SNL) to build sequencing software and/or connect custom C code to the real-time database.
5. Timestamping of data to 1usec accuracy. EPICS has been modified for LIGO to use the GPS VME modules as the time source. (Base EPICS uses time services from one processor (master) across ethernet to all other real-time processors.)

### 7.2.3.2 Design Analysis

#### *7.2.3.2.1 EPICS Capabilities*

EPICS consists of two primary parts, the core and extensions. Here the core is defined as the real-time components which provide the functions listed in the previous section. Extensions are typically Unix components, such as the HMI, described later.

The EPICS core is chosen as the control kernel for the following reasons:

1. Supported by a collaboration of labs and universities, primarily LANL, APS and CEBAF. The core has not changed notably over past year or so and has been fairly robust, as indicated by the success and popularity of the system.
2. Channel Access. This is a discovery protocol, which allows automatic lookup and connection to data residing anywhere on the network in any processor through the use of a unique name tag given each signal.
3. API. Channel access and real-time database calls are available for inclusion into custom software, allowing easy addition of new code and allowing code to make use of the EPICS core infrastructure.
4. Open system. This allows code to be added, as necessary, to meet particular LIGO needs.

#### *7.2.3.2.2 EPICS Limitations*

While CA is a major asset of EPICS, it also has been noted to have limitations which must be watched closely in design and implementation of the system. If network connections to a particular real-time processor running the CA server become more than it can handle i.e. CPU time is not available, it will arbitrarily discard data requested by CA clients. This can result in such things as:

- Two operator windows, which display the same data channels, not showing the same data values for those channels, because one was updated while the other's request was discarded.
- Real-time processes losing communication and synchronization.

Another drawback in the core for some LIGO applications is the processing overhead of the EPICS database entries, or records, each time they are updated. For example, an analog input record, which reads a single data value from a VME ADC, reads the value, performs engineering unit conversions, checks alarm limits, posts events, etc., and takes ~300usec on an MVME162 to process. Typical time for a single data value read in a standard C code call on this processor is < 2 usec. Therefore, EPICS database record processing is adding 298usec of overhead in its other processing. Where this is a problem is in the faster LIGO control loops, which need to run at 1KHz or better. Therefore, the general philosophy of building these control loops will be as was done in the LIGO PSL prototype, which is:

- Higher performance VxWorks supported processors, where necessary.
- EPICS records will be used primarily as entry points to communicate data through CA to other processors and operator interfaces rather than sequencing through a string of records to perform control functions.
- Custom C code will be written which performs the faster control algorithms and communicates directly to the VME I/O modules, rather than using the EPICS VME I/O capabilities.

Figure 22: PSL Prototype Control Software shows an implementation used on the prototype PSL. EPICS is used to handle slow (<10 Hz) control and monitoring functions. SNL based C code modules provide the higher speed (1KHz and greater) control, and only use EPICS to pass data to/from operators via the ethernet. Communication to/from VME modules is done directly by these modules and do not use the EPICS VME module drivers. In the future, these modules would also directly access the fiber channel network for high speed communication to/from other synchronized processes running in other VME crates

##### *7.2.3.2.3 Alternatives*

Most alternatives to EPICS which were identified are limited in one or more of the following areas:

1. Require proprietary hardware and bussing systems i.e. do not use an architecture with multi-vendor support, such as VME.
2. Do not provide high performance capabilities i.e. limited to 1Hz types of update rates.
3. Do not provide or have very limited networking capabilities.
4. Provide only a portion of the functionality required for a full control and monitoring system.

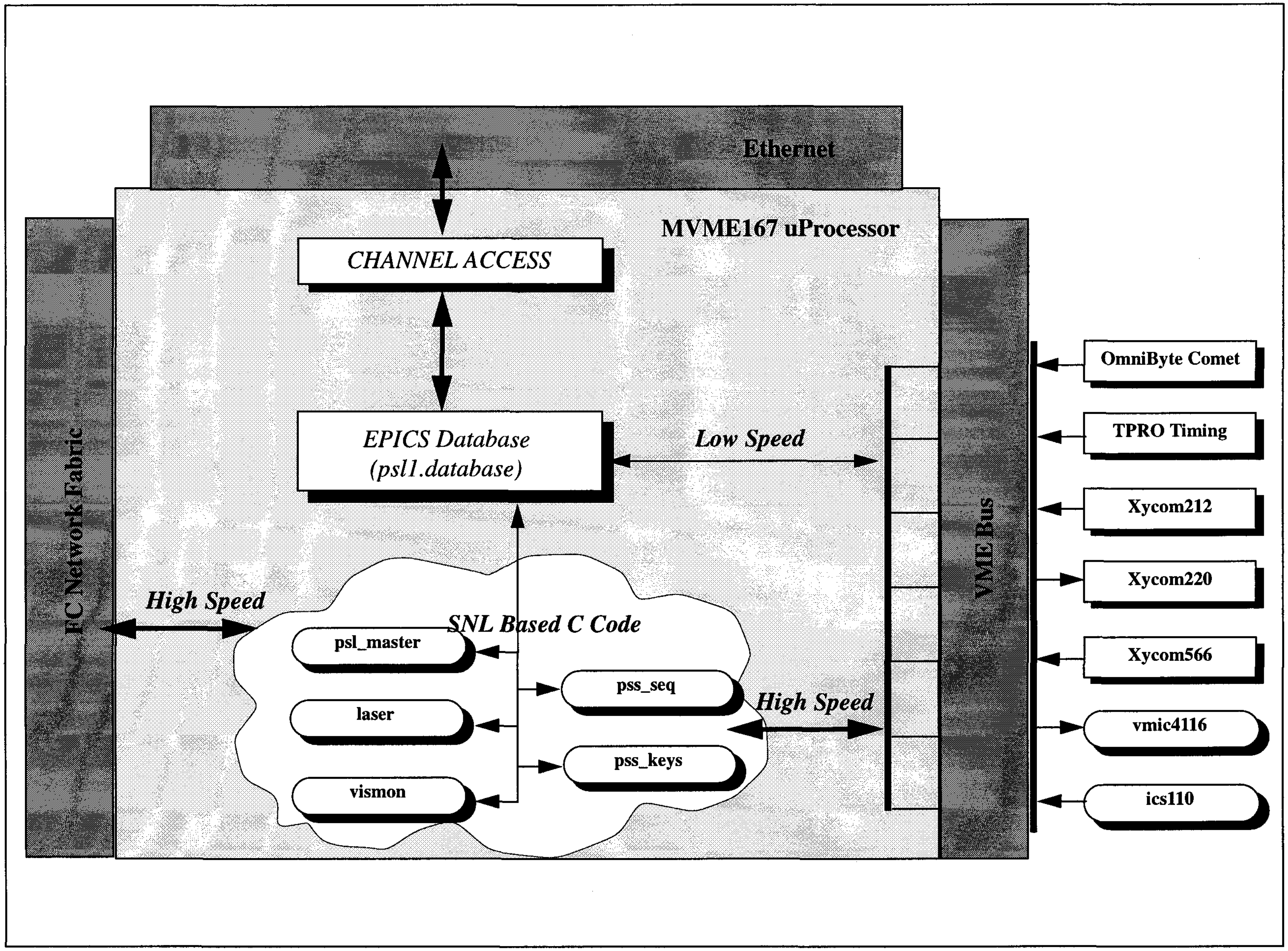



Figure 22: PSL Prototype Control Software

### 7.2.4. Development Tools

The software development tools to be used are shown in the upper left corner of Figure 21: Real-time Software Overview. As mentioned previously, the Tornado product from Wind River Systems will be used for C/C++ code development. A copy of Wind River System's world wide web page for Tornado is included in Appendix A for additional information.

For connection of CDS C/C++ software to the real-time EPICS database, the EPICS SNL code and pre-compiler will be used. This provides for standard calls to send/retrieve data to/from EPICS, as well as obtain monitors and events from EPICS.

To develop EPICS databases, CDS has adapted Microsoft Access. Several ascii and graphical editors already existed for EPICS, but they do not provide for all of the capabilities found in a commercial database, such as:

- Storing and relating other CDS information, such as hardware and wiring configurations, along with EPICS related fields
- Wide assortment of data queries.
- Varieties of report generation capabilities.
- Links, via commercial standards, to other software packages.

]

Adapting MS Access for use with EPICS was a fairly simple process, given the many tools provided in that product. The general layout and development flow is shown in Figure 23: MS Access to EPICS Database. The conversion between the MS Access database data format and the EPICS database format is handled by a CDS developed MS Access Basic post-processor. Refinements and additions will be made during the CDS preliminary design phase to provide additional functions necessary for code development.

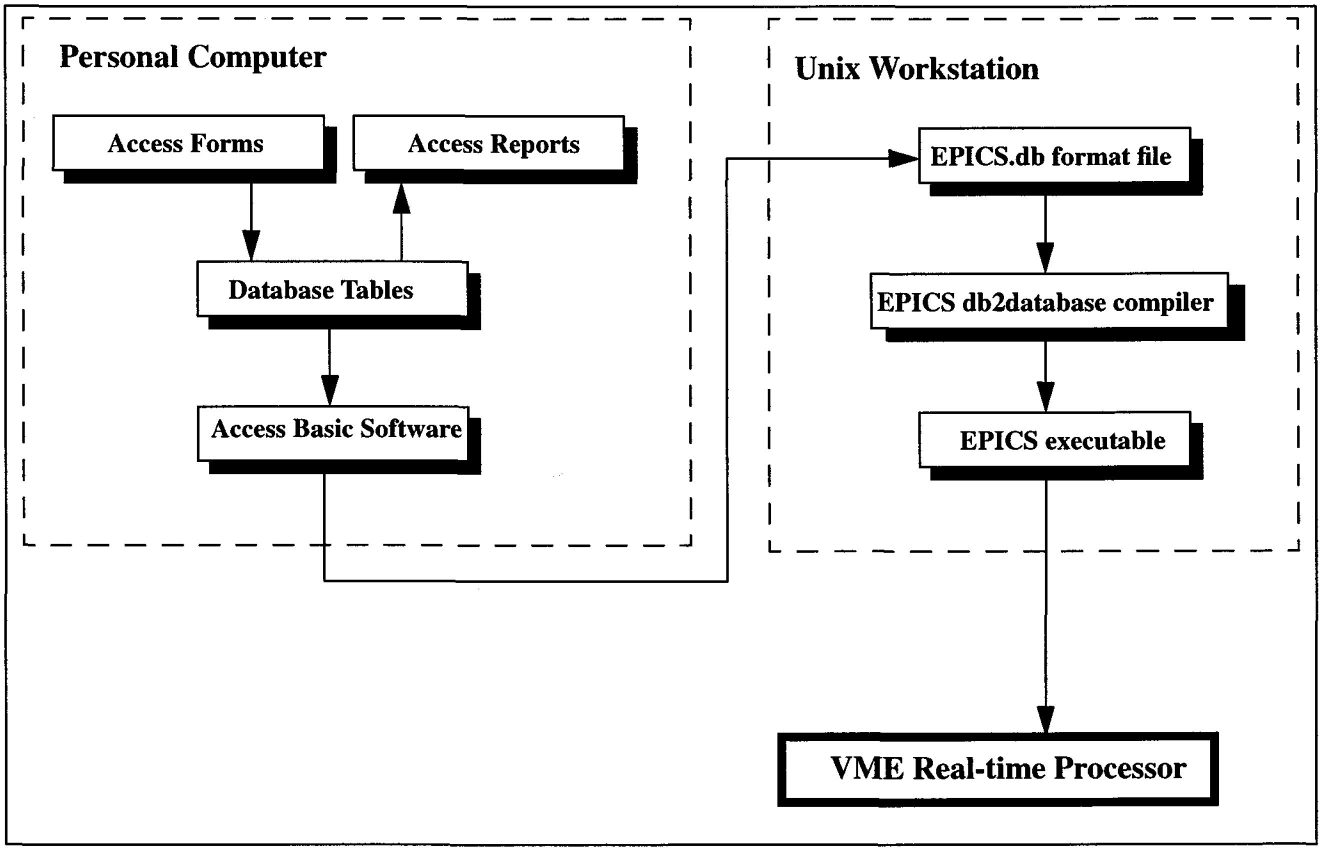



**Figure 23: MS Access to EPICS Database**

## 7.3. Non-real-time Software

### 7.3.1. Overview

In the case of CDS, the non-real-time category of software primarily includes:

- Inter-active HMI displays
- Alarm Management
- Data Archival
- System Diagnostics

These are shown in a block view in Figure 24: Non-real-time Software Overview and described in the following sections.

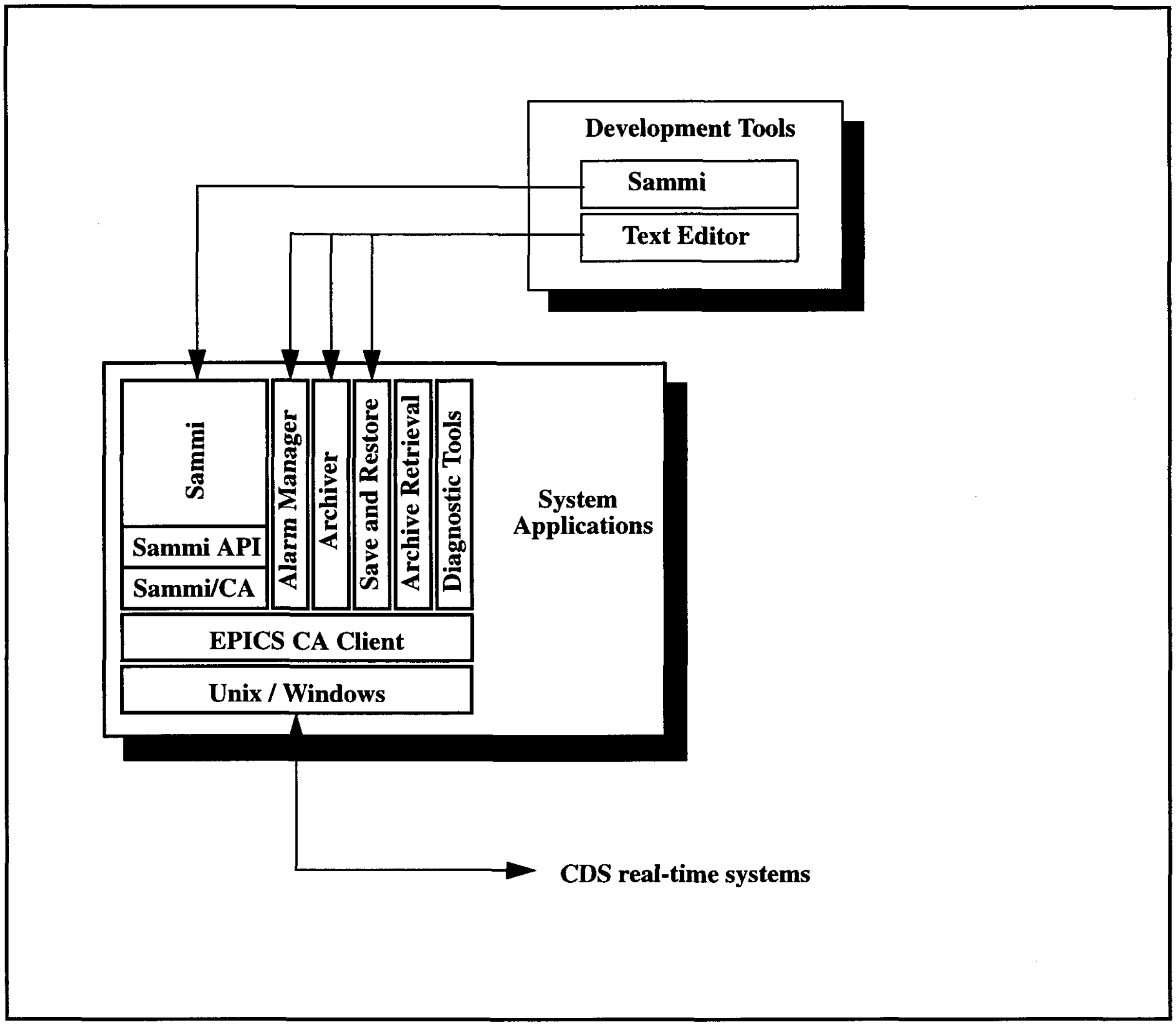



**Figure 24: Non-real-time Software Overview**

### 7.3.2. Operating Systems

The primary operating system for non-real-time applications will be Unix. Windows 95 platforms will also be supported for the HMI applications.

### 7.3.3. Control Kernel

EPICS will provide the control kernel, which, for non-real-time applications, primarily consists of the EPICS CA client. The CA client provides the communication mechanism to send/receive data to/from CDS front end real-time processors.

## 7.3.4. System Applications

### 7.3.4.1 Human-Machine Interface

#### *7.3.4.1.1 Design*

For purposes of both developing HMI and providing the interactive runtime HMI, the Sammi product from Kinesix will be used. A description of the Sammi product is given in Appendix B (copy of the Sammi brochure).

To make the connection with Sammi to the real-time data provided by EPICS, the Sammi API and a custom API to EPICS will be developed and connected as shown in Figure 25: HMI to CDS Real-time Connection. The basic sequence to build, run and connect HMI will be as follows:

1. GUI is built using Sammi widgets. Dynamic widgets are related to real-time data by entering the signal tag name to be controlled/monitored.
2. The Sammi development tool creates a binary file based on the screen built by the developer.
3. Sammi runtime is started on an operator console, and the desired HMI screen(s) brought up. These screens can run in X windows or on Windows 95 (primary operator consoles will be X windows, whereas remote, maintenance units will be laptop PCs with Windows 95). Sammi runtime will connect with a Sammi API running on a CDS server. This is done via a standard RPC connection, supported within Sammi.
4. The Sammi API connects via a pipe to an EPICS client (to be developed) within the console server, which translates Sammi requests to EPICS CA calls.
5. The EPICS client finds and connects the data requests to CDS real-time processors.
6. With data connections made, the HMI is now interactive with the operator, providing real-time data updates, and responding to operator inputs/commands.

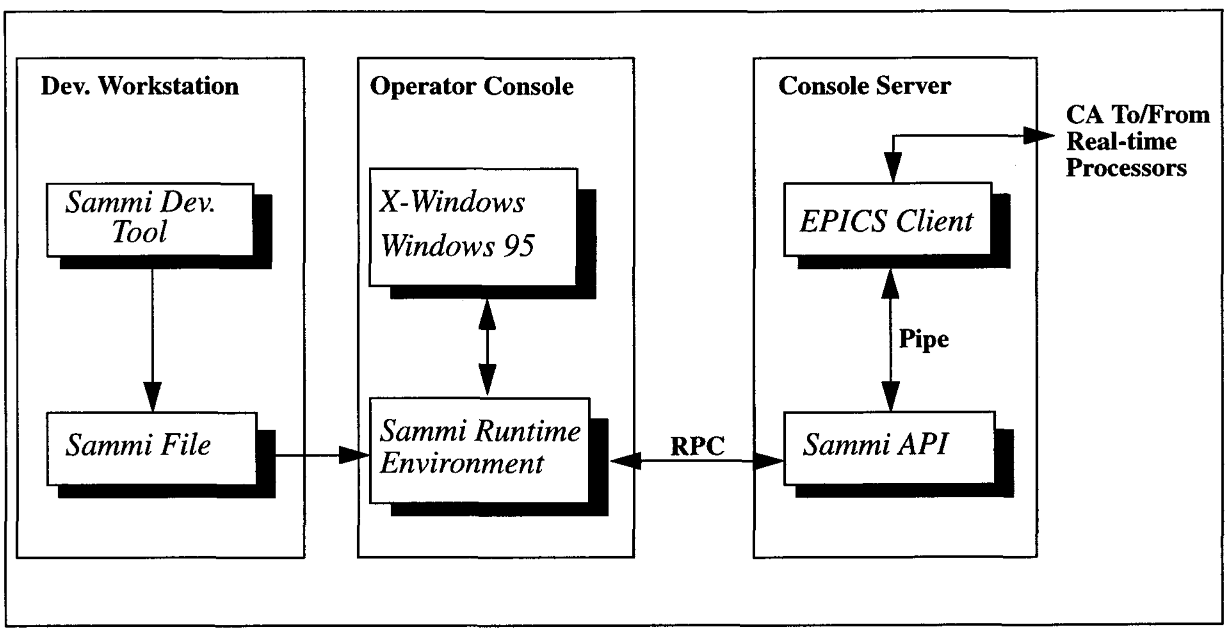



**Figure 25: HMI to CDS Real-time Connection**

#### *7.3.4.1.2 Design Analysis*

Several HMI development and runtime tools were investigated. The first approach was to use the EPICS supported extensions, Motif Editor and Display Manager (MEDM) and the original EPICS edd/dm. Problems encountered with these are:

1. MEDM:
   - Solaris version tends to crash frequently, which is unacceptable.
   - Sun OS version fairly stable, but it is not desirable to stay with the older version of Sun operating systems.
   - Only one member of EPICS collaboration familiar enough to make necessary fixes and/or upgrades, and is not presently working on system.
   - Overall, MEDM seems to be a dead end project within the EPICS collaboration.
2. edd/dm:
   - Poor overall appearance when compared with MEDM or any commercial product.
   - Collaboration wants to upgrade this to be MOTIF-like, but needs infusion of about $100K to pay for the work.
   - Quick glance of the beginnings of MOTIF-like edd/dm shows signs of bugs and poor appearance.

With the problems of the EPICS built-in extensions, several commercial packages haves been investigated. Along with Sammi, evaluated were SL-GMS and UIM/X with XRT. All have strong customer bases in the GUI development market. Further evaluation will be done during the preliminary design phase, but Sammi appears superior in:

1. Only package of the group designed specifically for control system HMI and has most of the standard widget libraries built in.
2. Contains tools to develop further widgets which are then added to the Sammi library.
3. Operates stand-alone:
   - Displays are built totally through point and click graphics and do not require embedding code and then compiling the runtime display. In contrast, UIM/X and XRT provide point and click graphics to layout an HMI, but this only produces skeleton code (which handles the widgets drawn), and requires the user to add all of the necessary call back code on what actions to take to move data from/to the widgets.
   - Allows users other than programmers the ability to develop HMI as no coding is required.
   - CA interface to EPICS can be made stand-alone, thereby not requiring recoding of screens if EPICS code changes or Sammi software changes.
4. Fault tolerance with automatic switch over to alternate data servers.

### 7.3.4.2 Data Archival and Retrieval

#### *7.3.4.2.1 Data Archival*

Data archival for the Control and Monitoring portion of CDS consists of archive to disk/tape of:

1. Slow data (10 Hz or slower), recorded for purposes of analyzing trends in LIGO operations.
2. High speed (up to 20M samples/sec) snapshot data used to analyze servo loop systems.

The archival of continuous high speed data (up to 6MBytes/sec) will be handled by a separate CDS data acquisition system, described in TBD.

To perform these functions, the EPICS ARchiver (AR) extension will be used as the base, with LIGO extensions to be added. The archiver software arrangement is shown in Figure 26: Data Archive Software Structure.

To archive data, the user first develops an ASCII file, which denotes the data channels to be archived, how often, and to what file system. This file is then loaded to the AR runtime, which connects, via CA, to real-time data channels and performs the actual archive process.

AR has two limitations which will need to be addressed to meet LIGO requirements:

1. All data described by an ASCII file is archived to a single file. While this is satisfactory for short periods of data storage (several hours), data which is archived continuously over days takes an unsatisfactory amount of time (5 minutes or more when only a dozen channels are being archived over a 24 hr period once every 10 seconds) to retrieve from the AR format. The intent to correct this is to post process data files in a manner TBD. This has been a standing recommendation within the EPICS collaboration, but no one has stepped forward to undertake the task.
2. AR will not archive array data, such as would be the format of the high speed snapshot data. To accomplish this, either an extension will be made to AR or a separate CA client process built.

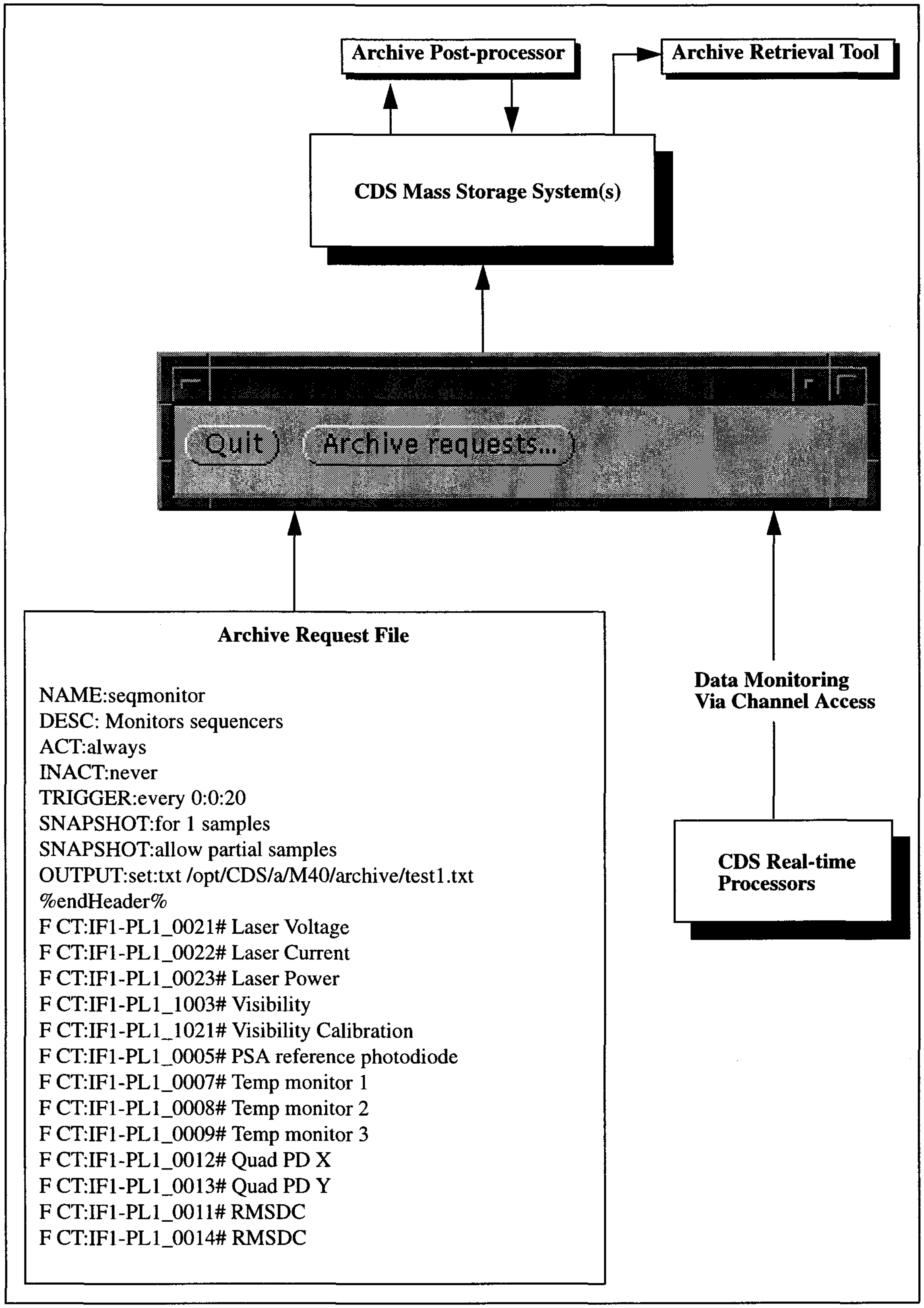



**Figure 26: Data Archive Software Structure**

#### *7.3.4.2.2 Archive Data Retrieval*

##### *7.3.4.2.2.1 Design*

Data retrieval will be via the latest EPICS ARchive Retrieval (ARR) tool extension developed using tcl/tk. An ARR interface screen is shown in Figure 27: Archive Retrieval Display Example. Features of ARR include:

- Point and click GUI
- Graphic and Tabular data representations.
- Multi-parameter, multi-scale, 2D plotting with zoom, pan, autoscaling, legends.
- Postscript printouts

##### *7.3.4.2.2.2 Design Analysis*

The latest EPICS ARR provides superior display representation capabilities to the previous versions, though its hardcopy is somewhat inferior. It meets most of the LIGO requirements and needs, except needs improvement in:

- Data retrieval time. Data retrieval for long files (12+ hours) takes an uncomfortable amount of time to retrieve (as previously mentioned). This is due to a combination of the file formats and the ARR code itself.
- Scaling. Plots can only be on single scale or auto scale all. There is no feature for individually scaling multiple plots.
- Tool can only retrieve and plot data from one archive file at a time (no ability to call up and overlay data from multiple archive files).

To address these shortcomings, either ARR will be enhanced directly, or Sammi or AVS will be used to replace it in the long term. Sammi and AVS already have built in features to handle these data retrieval and plotting needs, but will need an interface built to the data (which should be straightforward).

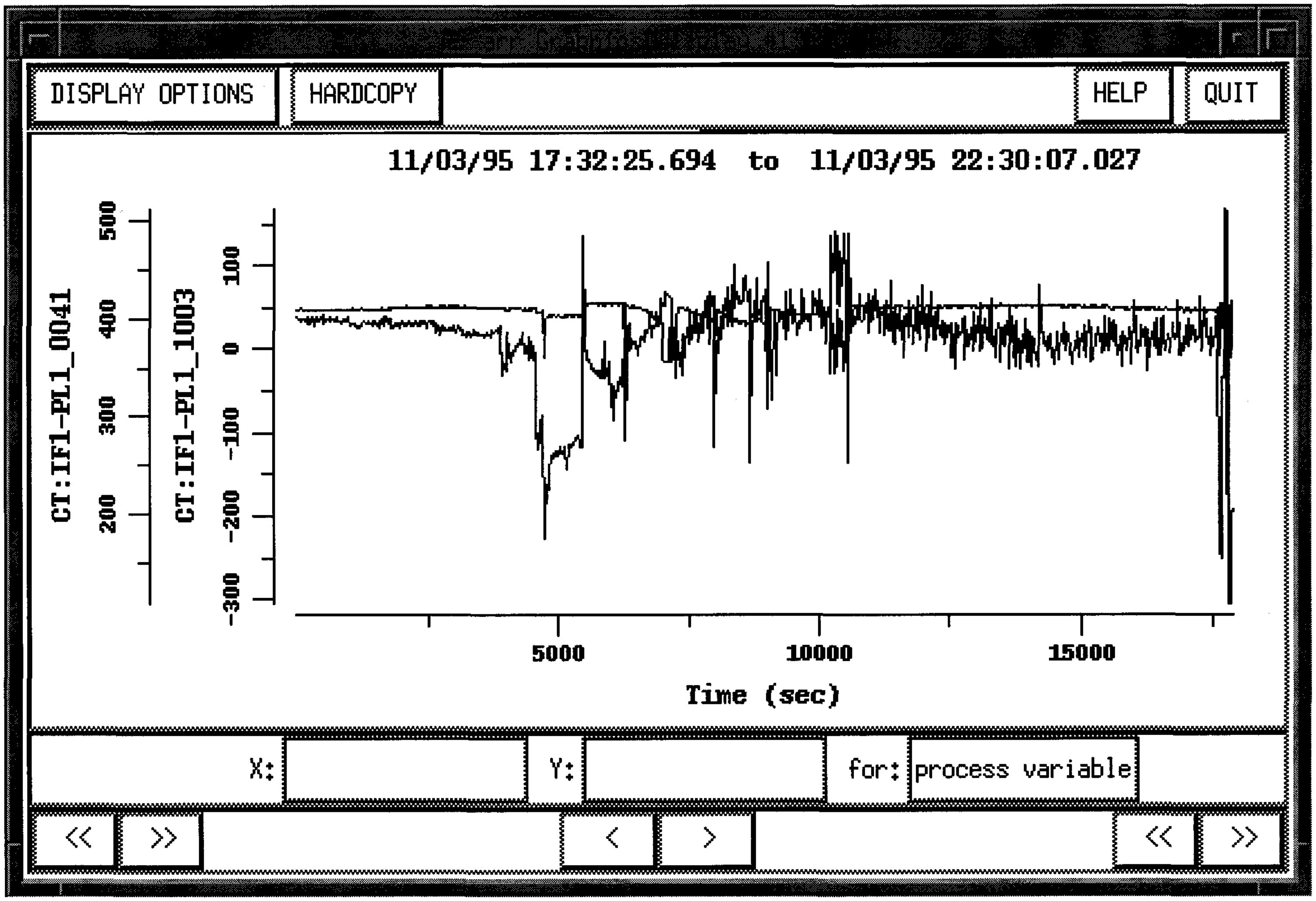



**Figure 27: Archive Retrieval Display Example**

#### 7.3.4.3 Alarm Management

Alarm enunciation, display and logging will be provided using the EPICS alarm manager (ALM). An overview of ALM is shown in Figure 28: Alarm Management Software Architecture. ALM allows for:

- The definition and structuring of alarm trees via an ascii editor using ALM keywords and guidelines.
- Alarm enunciation and display of the alarm tree as shown in Figure 29: Alarm Manager Example.
- Alarm logging and playback as shown in Figure 30: Alarm Log File Example.
- Defining and displaying operator guidance along with the alarm states.
- Defining and allowing operator execution of real-time processes to deal with alarm conditions.

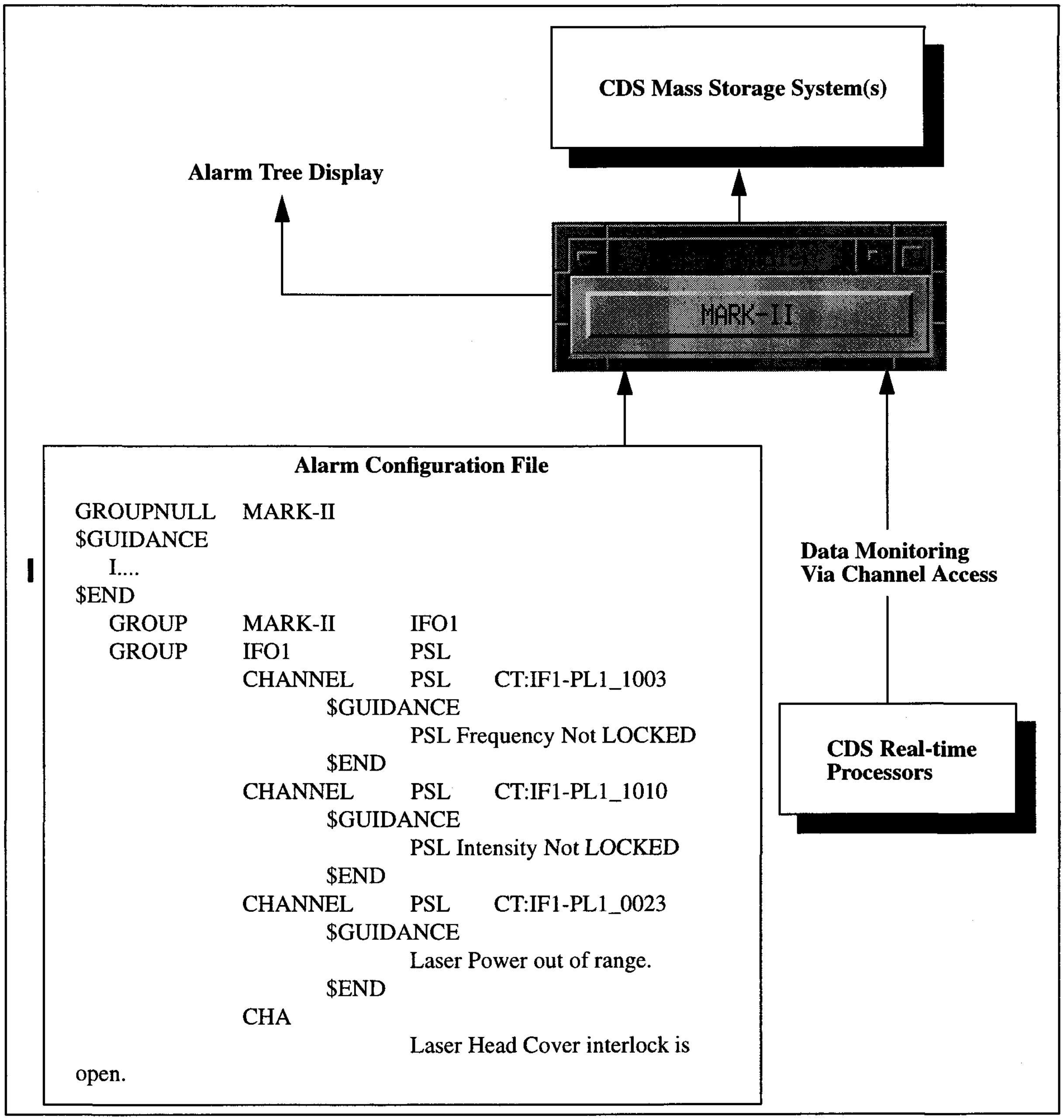



**Figure 28: Alarm Management Software Architecture**

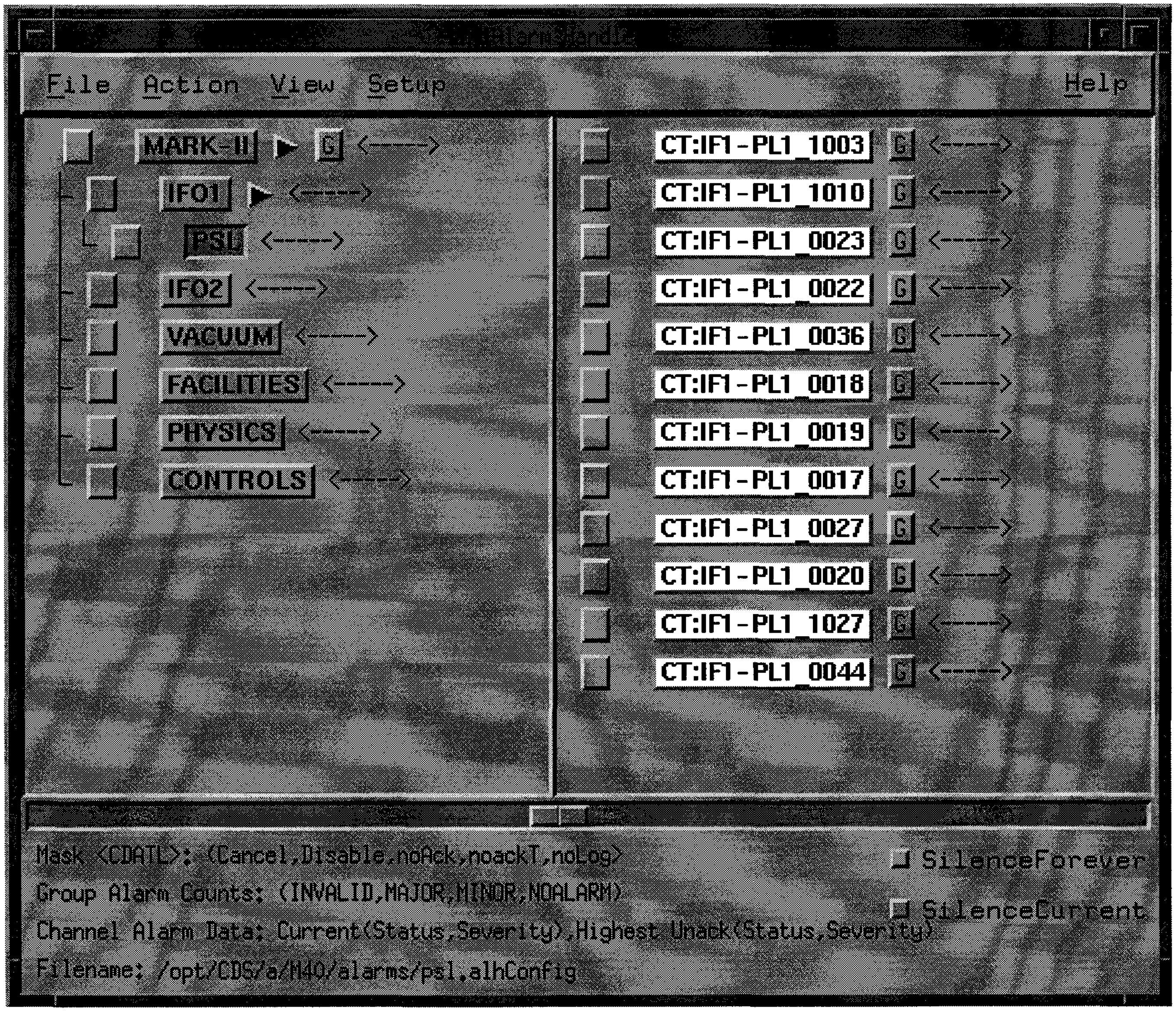



**Figure 29: Alarm Manager Example**

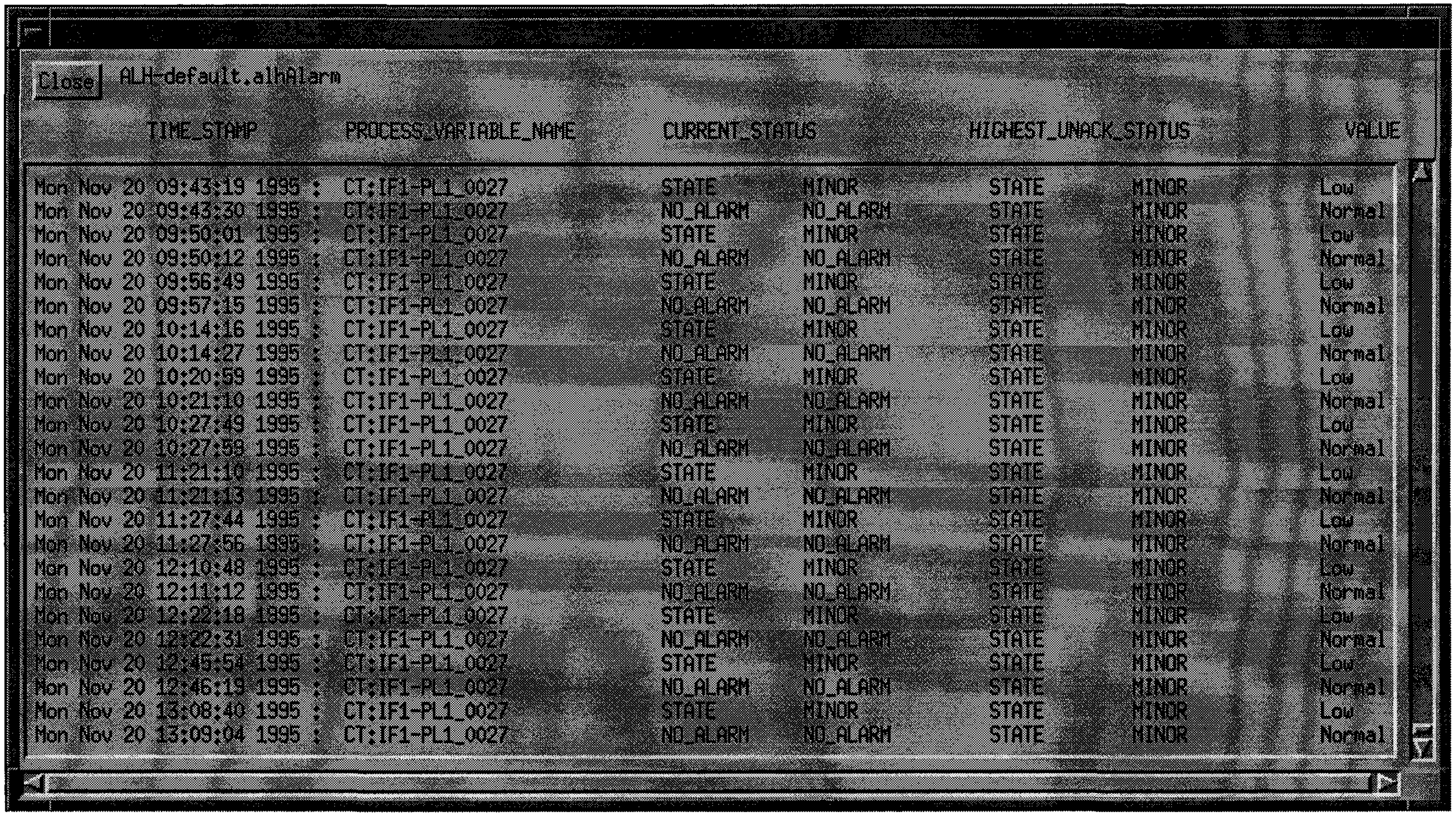



**Figure 30: Alarm Log File Example**

#### 7.3.4.4 Save and Restore

Save and restore provides the capability to take "snapshots" of CDS control settings/readings to allow resetting control parameters to the same configuration at a later time. The Back-Up and Restore Tool (BURT) of EPICS will be used to provide this functionality. Figure 31: Backup and Restore Tool (BURT) depicts the user interface windows.

BURT provides:

- Collection and storage to user defined files of system setpoints and readings. Data to collect is defined by the user in ascii files using BURT keywords and structures.
- Resetting of setpoint parameters on demand from the operator.
- Viewing and modification capabilities prior to resetting values.
- Concatenation of multiple back-up files.
- Basic math routines to adjust back-up settings prior to resetting the real-time systems.

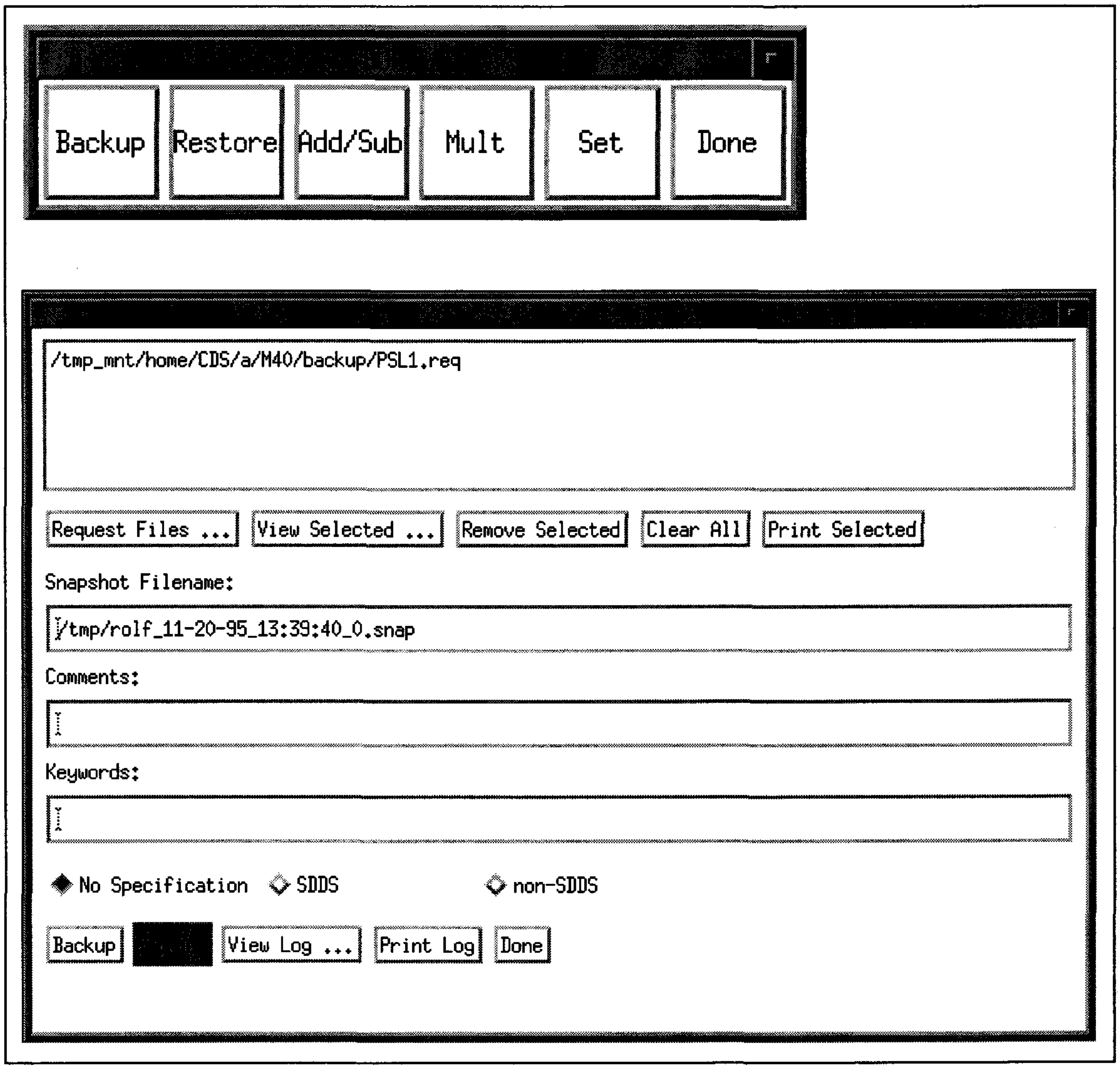



**Figure 31: Backup and Restore Tool (BURT)**

#### 7.3.4.5 System Diagnostics

The initial set of diagnostics will be those provided with the VxWorks and Unix operating systems, along with the EPICS tools. The EPICS tools include:

VxWorks command line interrogation, such as listing of records, records attached by SNL code, and status of I/O drivers.

Probe: An X window tool which provides display of values from EPICS record fields.

In the longer term, GUI interfaces will be built onto the system to provide:

- Status of all CDS software modules.
- Status of all CDS I/O modules.
- Status of all CDS networks.
- Status of all CDS mass storage systems.

### 7.3.5. Application Programmer's Interface

To provide connection of CDS data to code developed by other user's the primary API will be the CA call libraries and the EZCA libraries. Both provide embeddable C calls to allow access to EPICS data via CA. CA libraries provide the most versatility in asynchronous callbacks, but require a higher level of programming skills. EZCA provides easier to use function calls, but is more limited in its capabilities. Instructions for use of these libraries are provided in the EPICS manual set.

### 7.3.6. Information eXchange Services (IXS)

Data stored by the CDS will be in CDS defined formats. TBD data format translation routines will be provided by CDS to export stored data to other systems.

# Appendix A: Tornado/VxWorks Information (Wind River Systems)

# *Appendix B: Sammi GUI Software Information (Kinesix)*