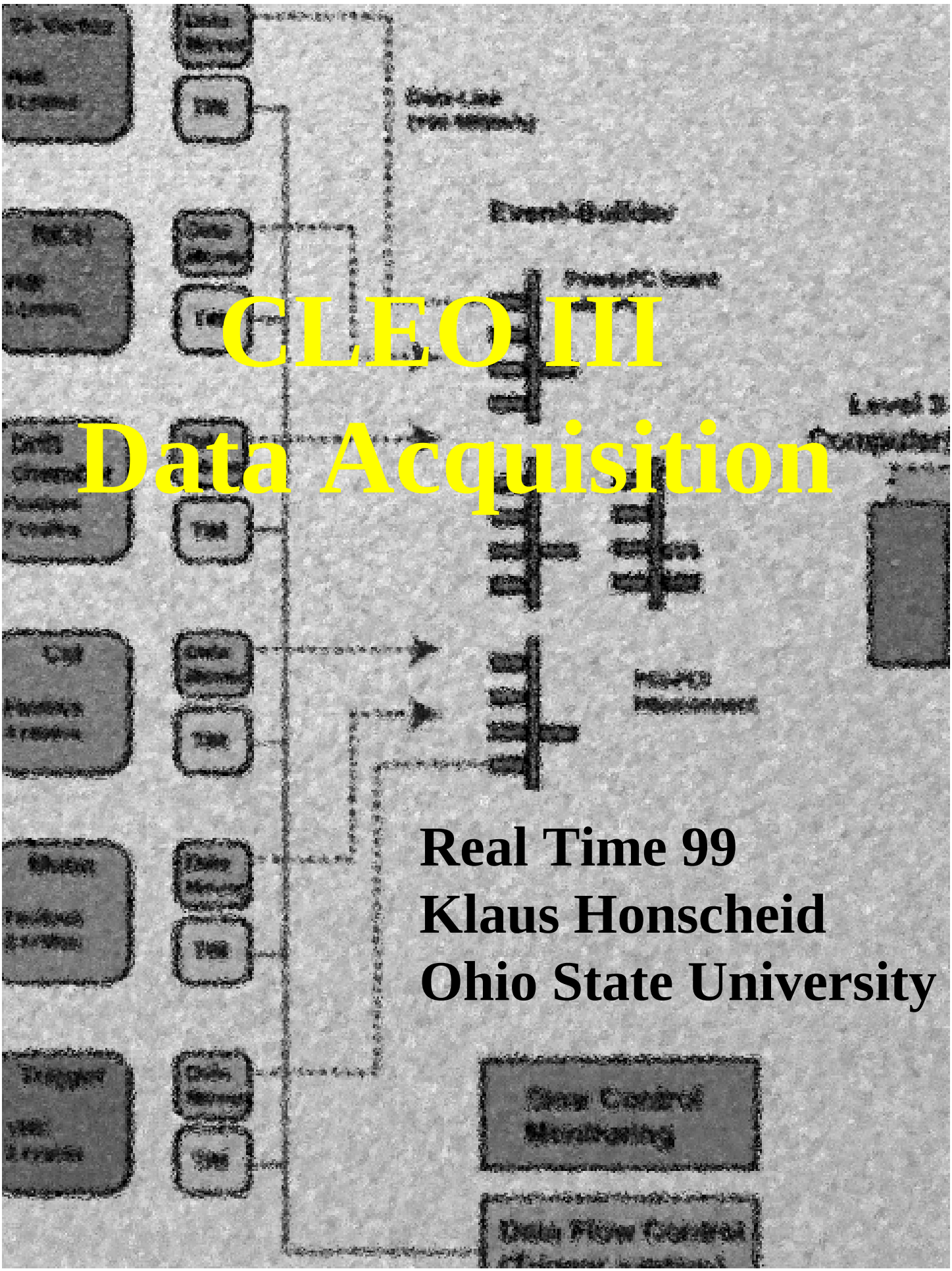
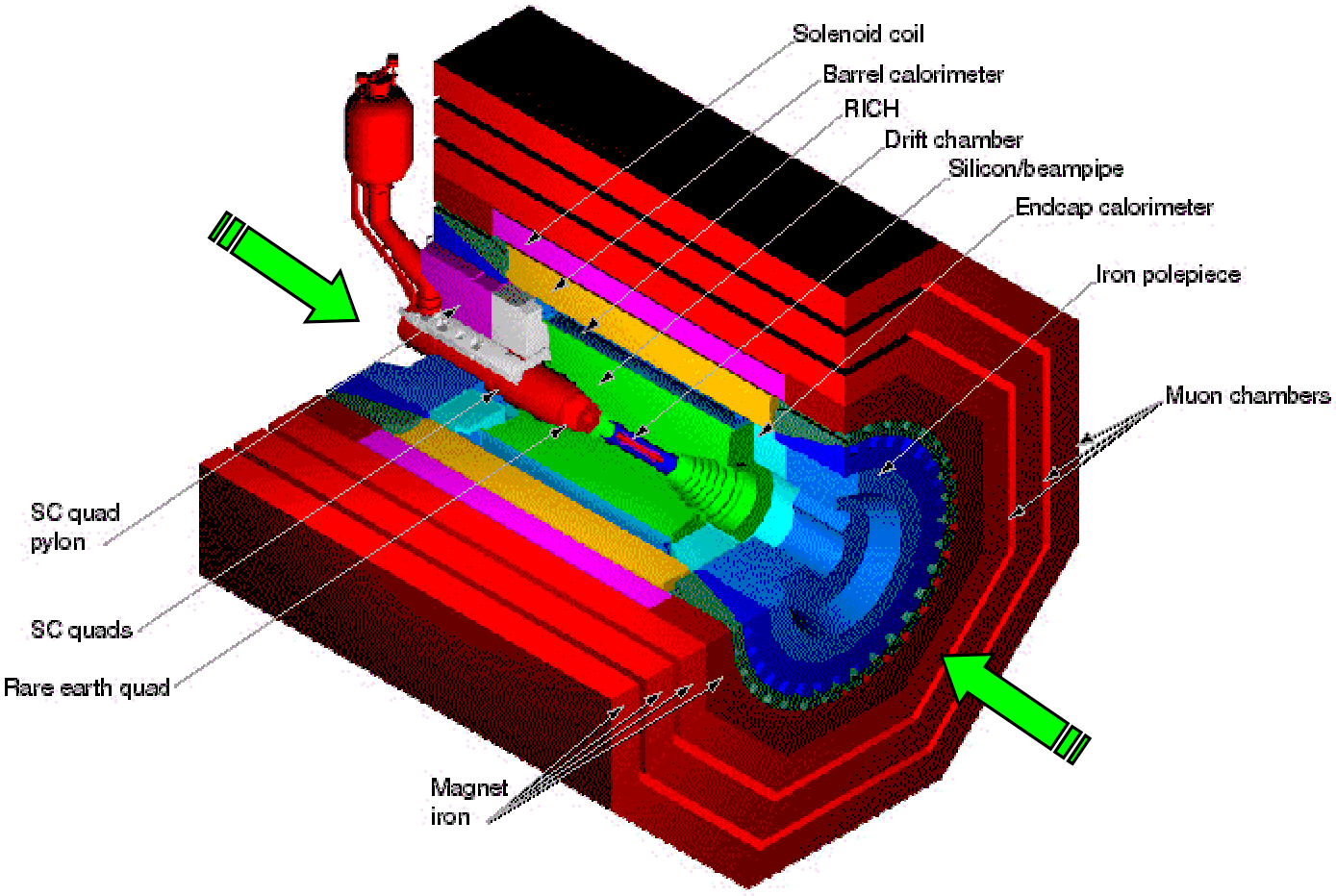


CLEO III Data Acquisition

Real Time 99

Klaus Honscheid
Ohio State University

CLEO III



CLEO Physics Goals:

- **Precision Tests of Standard Model**
- **Search for Rare Processes**

Both Require:

- **Higher Data Rate**
- **Sophisticated Control and Monitoring**

RT 95 (Michigan State): The Plan

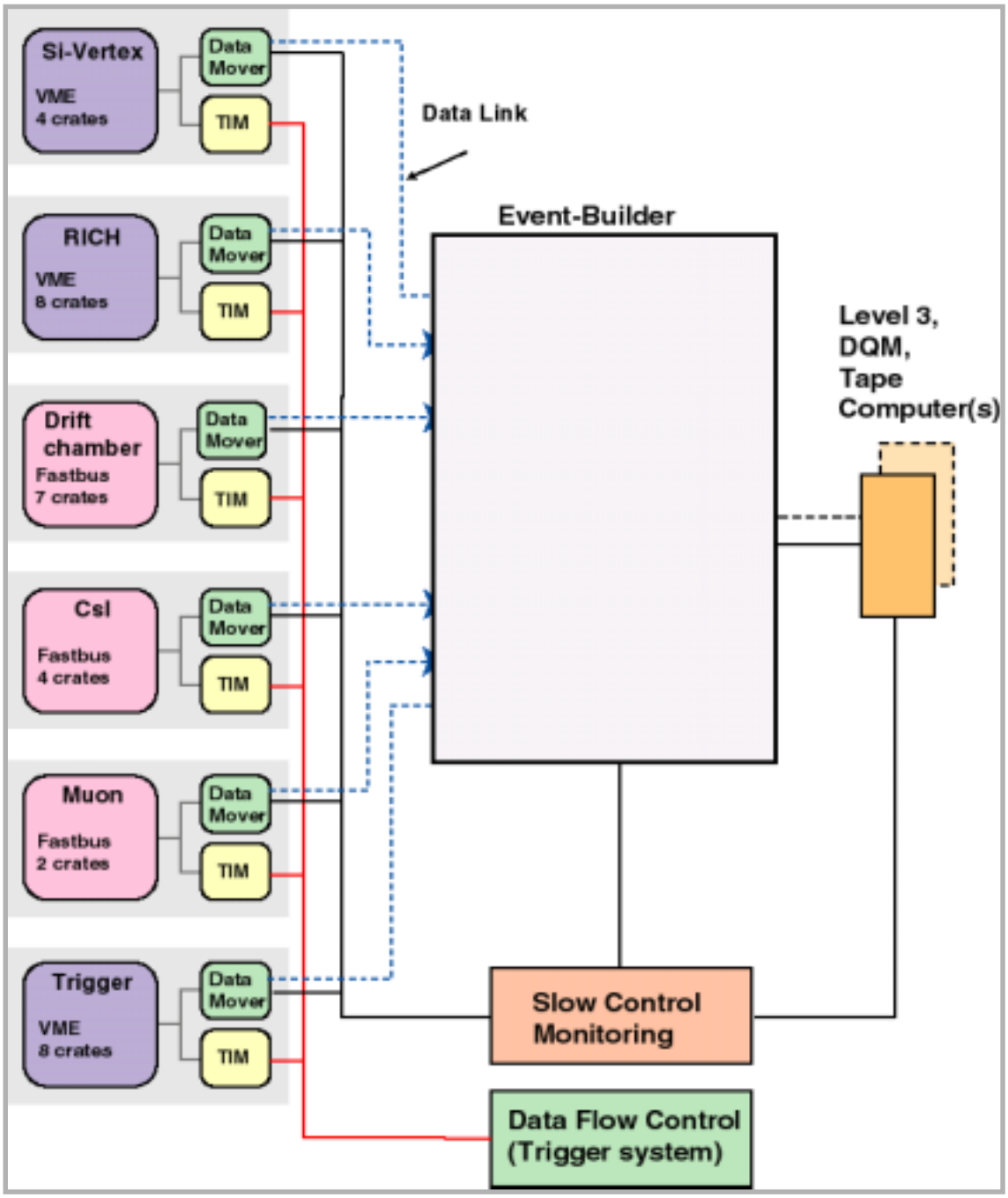
CLEO III Data Acquisition

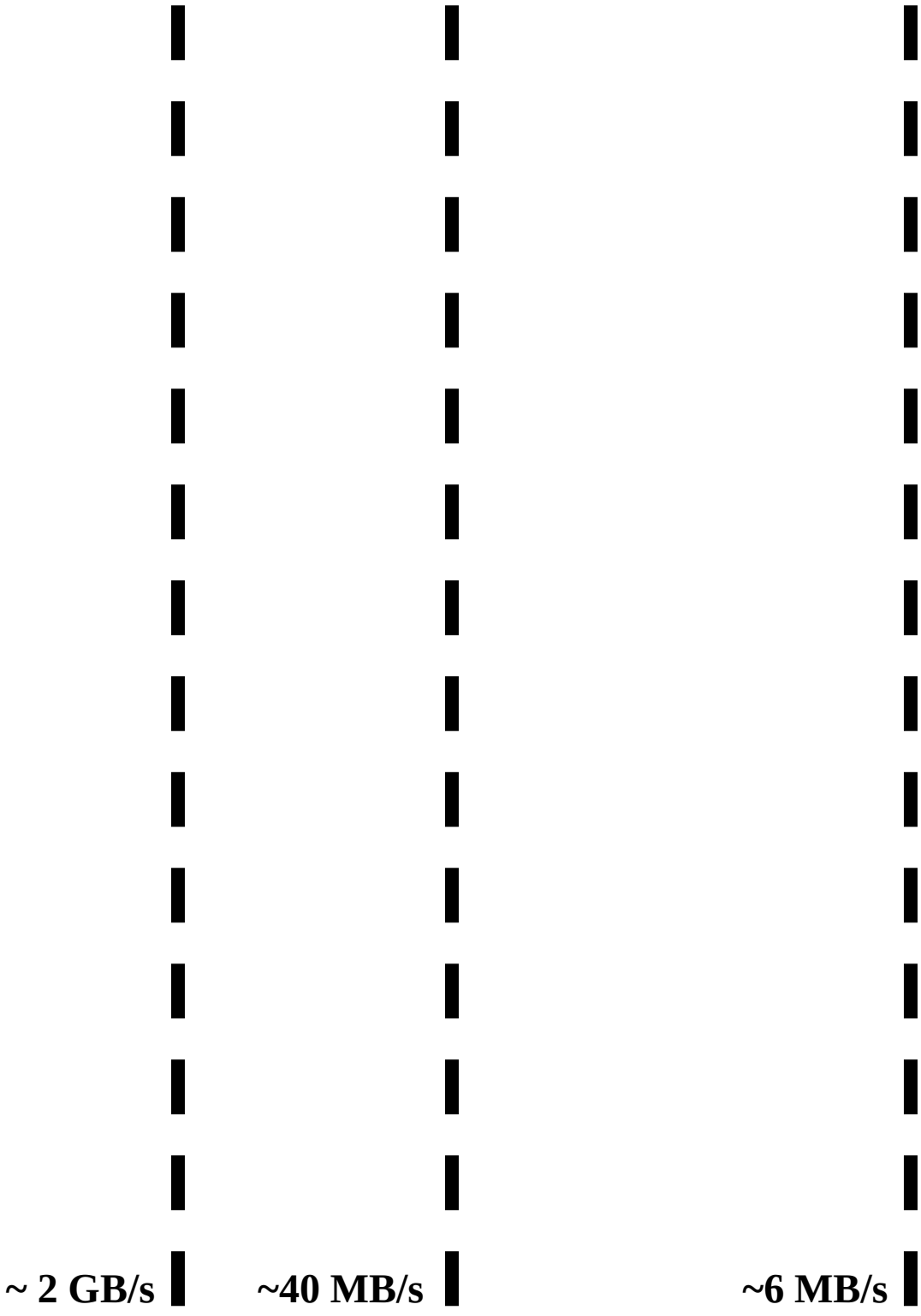
- Trigger Rate 1000 Hz, 25-50 KB/event
- keep it simple
- Using "standards" most of the hardware can be bought (later!)
- 3 hardware projects underway
  - » **FRITZ VME-FB Interface**
  - » **Databoard Buffer Manager**
  - » **Optical PCI-PCI/PMC Link**
- Modsim simulation
- Object oriented approach to Slow Control (CORBA)

  Done in 2 years (?)

CLEO III DAQ Architecture

- 20 μs readout time, up to 1000 Hz trigger rate
- ~40 Mbytes/s throughput




~ 2 GB/s
~40 MB/s
~6 MB/s

Eventbuilder Evolution

In the Beginning (RT 95)

- Custom Data Links
- Barrel Shifter Eventbuilder
- Custom Link to Computer (Farm)
- Budget: ???

Use Standards

- ATM Data Links
- ATM Switch (48 ports)
- ATM Link
- Budget a few 100 K$

Let's be realistic (RT 97)

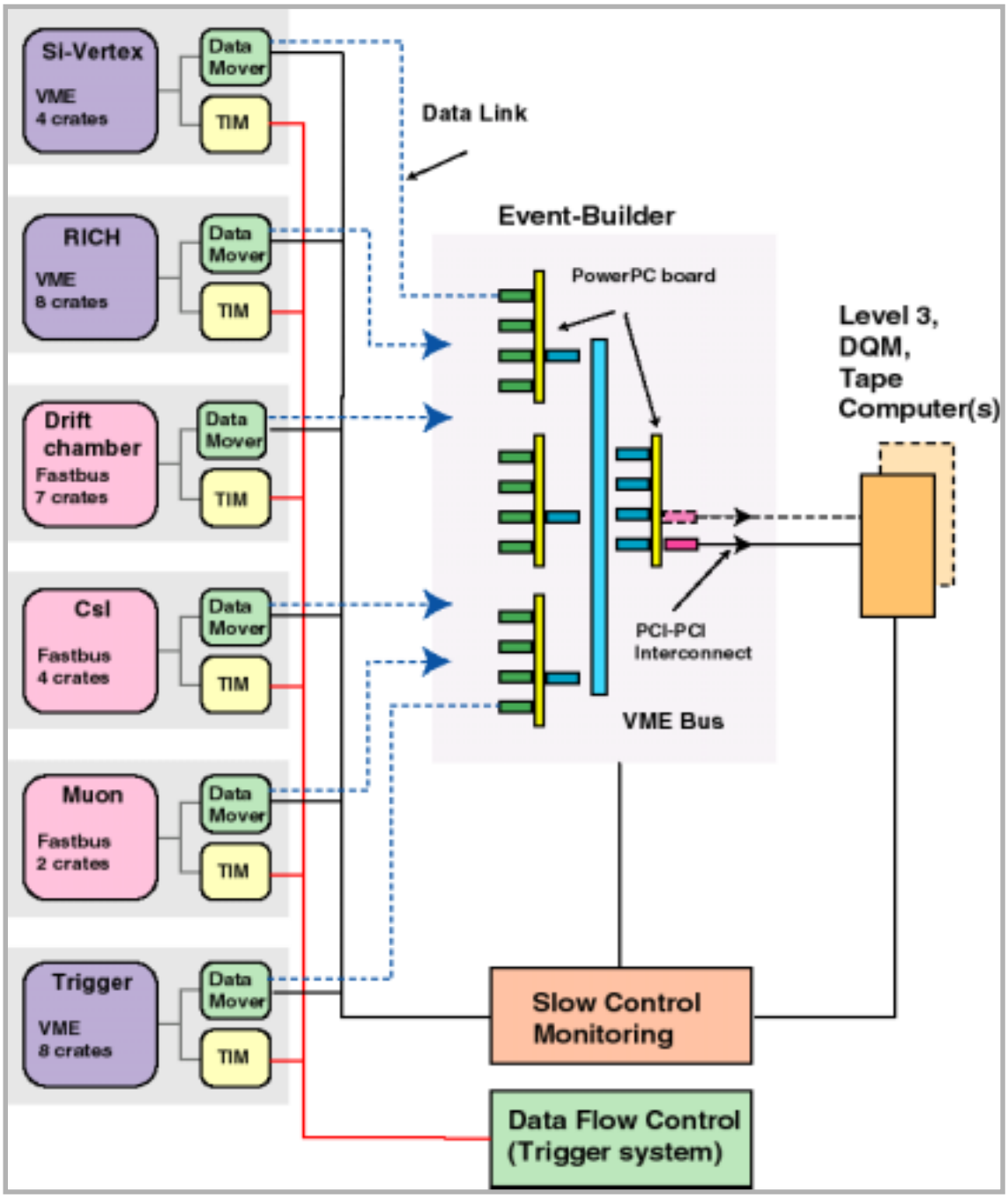
- Fast Ethernet Data Links
- Multi VME CPU, shared memory Eventbuilder
- Custom Link to Computer (Farm)
- Budget ~ 100 K$

Even simpler (RT 99)

- Fast Ethernet Data Links
- Fast Ethernet Switch + Solaris Computer
- Fast Ethernet Link to DQM
- Budget ~ 20 K$

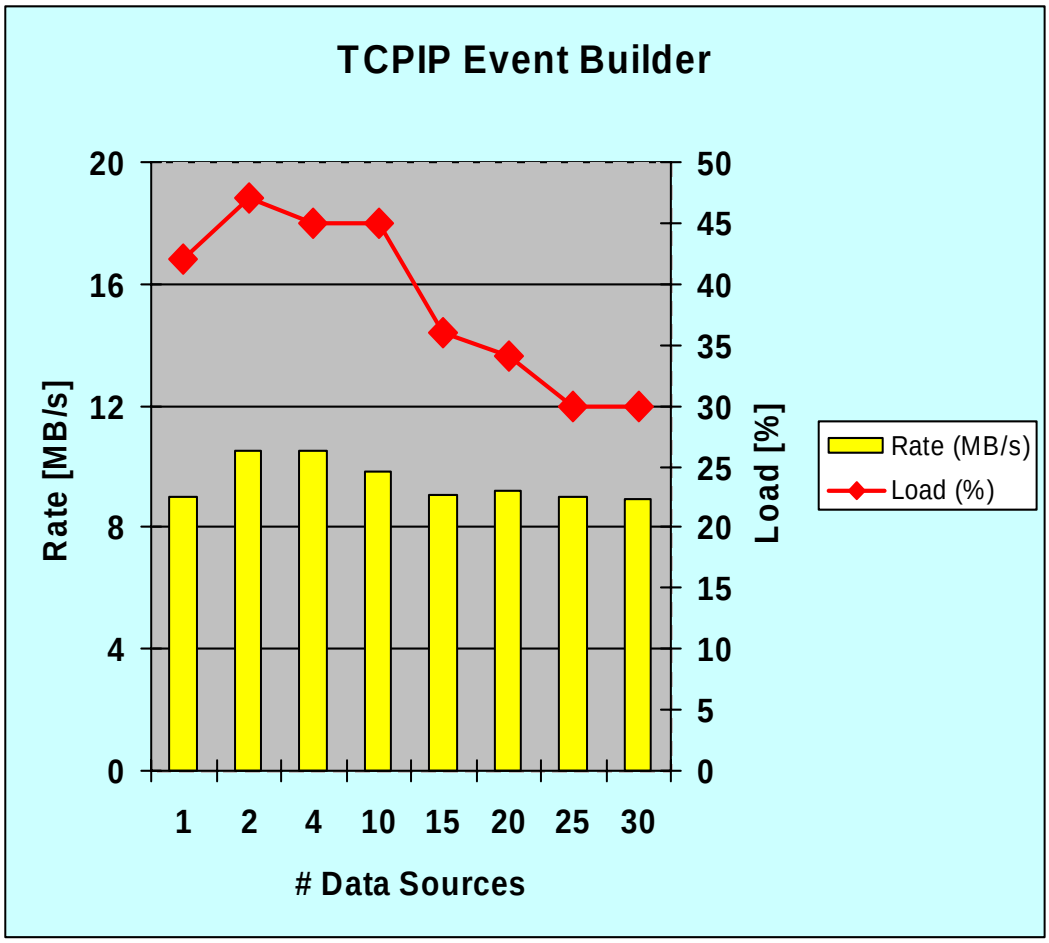
CLEO III DAQ Architecture

- 20 μs readout time, up to 1000 Hz trigger rate
- ~40 Mbytes/s throughput



Test:

- MV2306, Win NT4, Unix data sources
- HP Fast Ethernet Switch
- Ultra Sparc 5 Eventbuilder



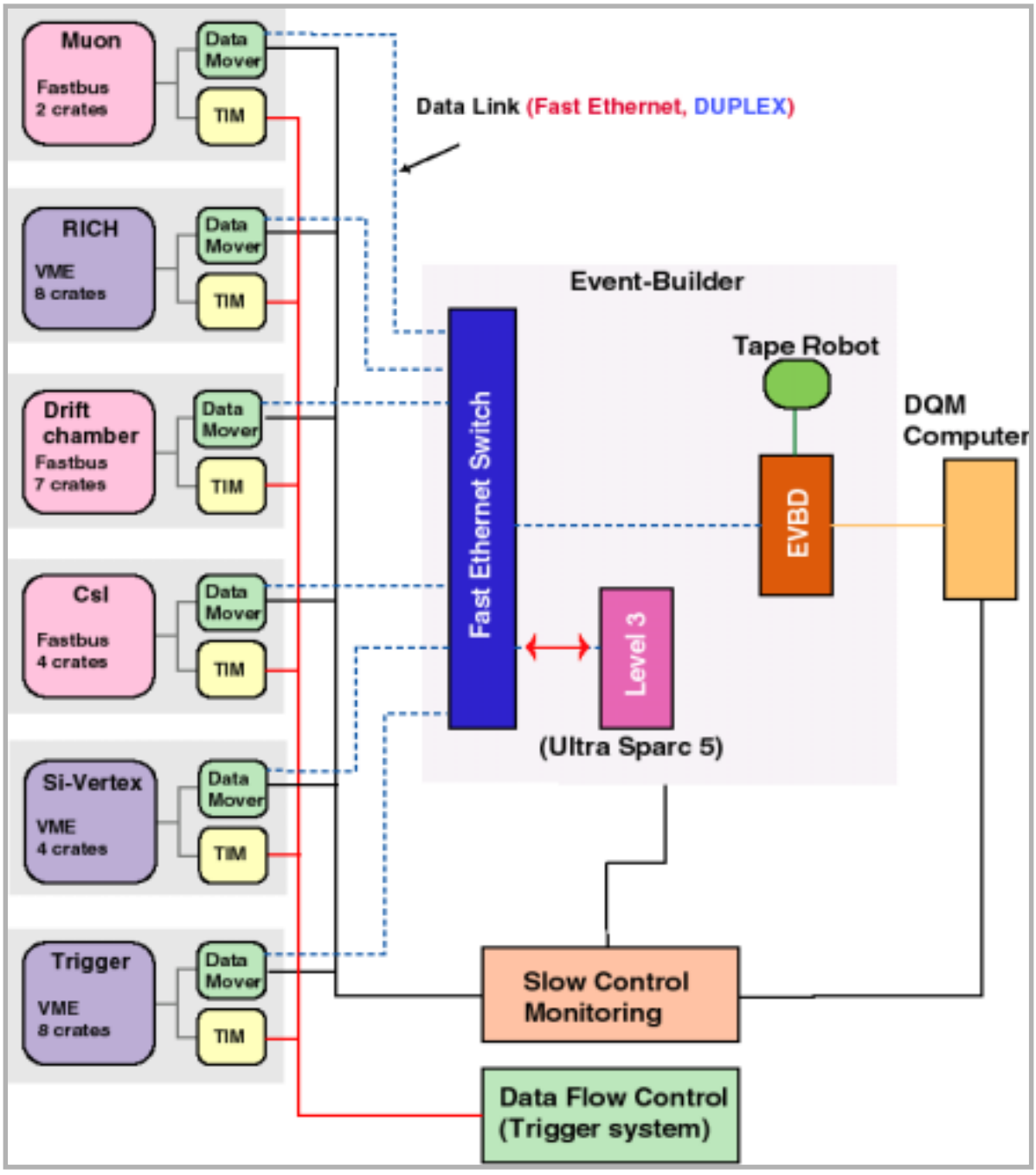
Benefits:

- Free Backpressure / Dataflow Control (TCP/IP)
- Free Data Link (Built in Fast Ethernet)
- Solaris vs. vxWorks development

But 9 MB/s is not enough - or is it?

CLEO III DAQ Architecture 3

- **Partial event data to Level 3 Processor (~1 MB/s)**
- **Reduced rate to Event Builder (~6 MB/s)**



Readout Controller

VME

Solution: PowerPC + VxWorks

Let Motorola worry about improvements…

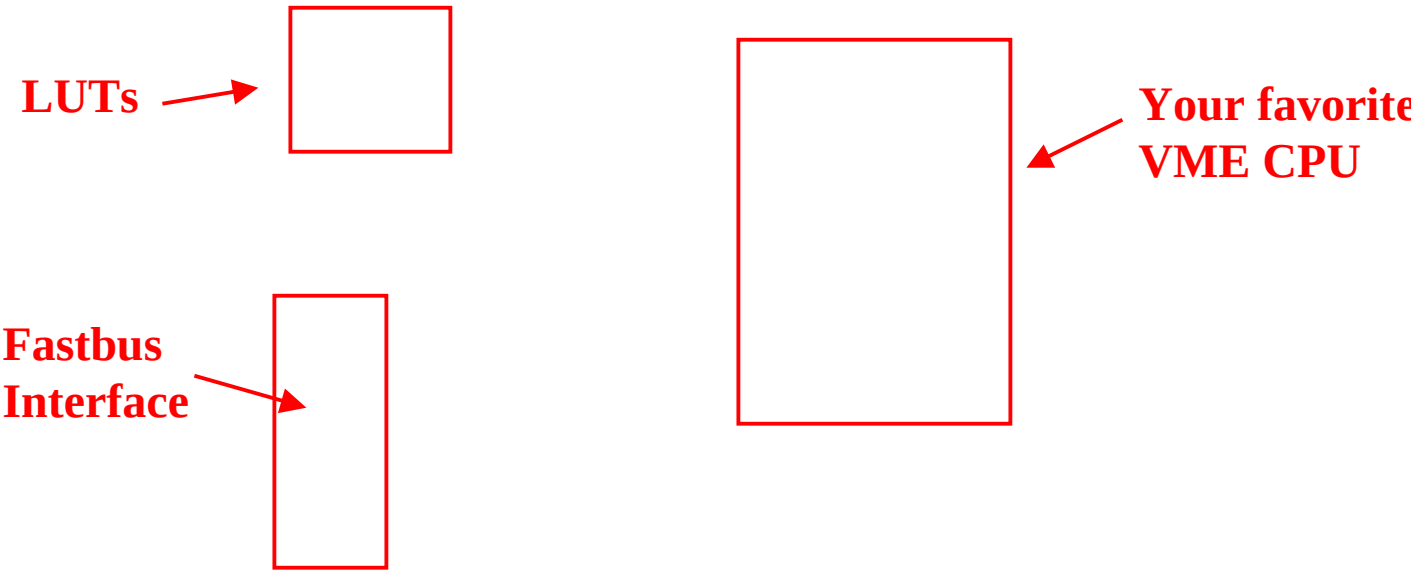
1995: mv1600
1997: mv2600
1998: mv2300
1999: mv2400

Price ⬇ Performance ⬆

Fastbus

Not much of a market!

Solution: Design own VME-Fastbus Interface (FRITZ)



Experience with CLEO III

Hardware

Use standards

(Bus system, Network protocol …)

Avoid custom hardware

(exceptions confirm the rule)

Don’t lock in too early

Don’t underestimate the market

(Network speed, CPU performance …)

Vendor Support

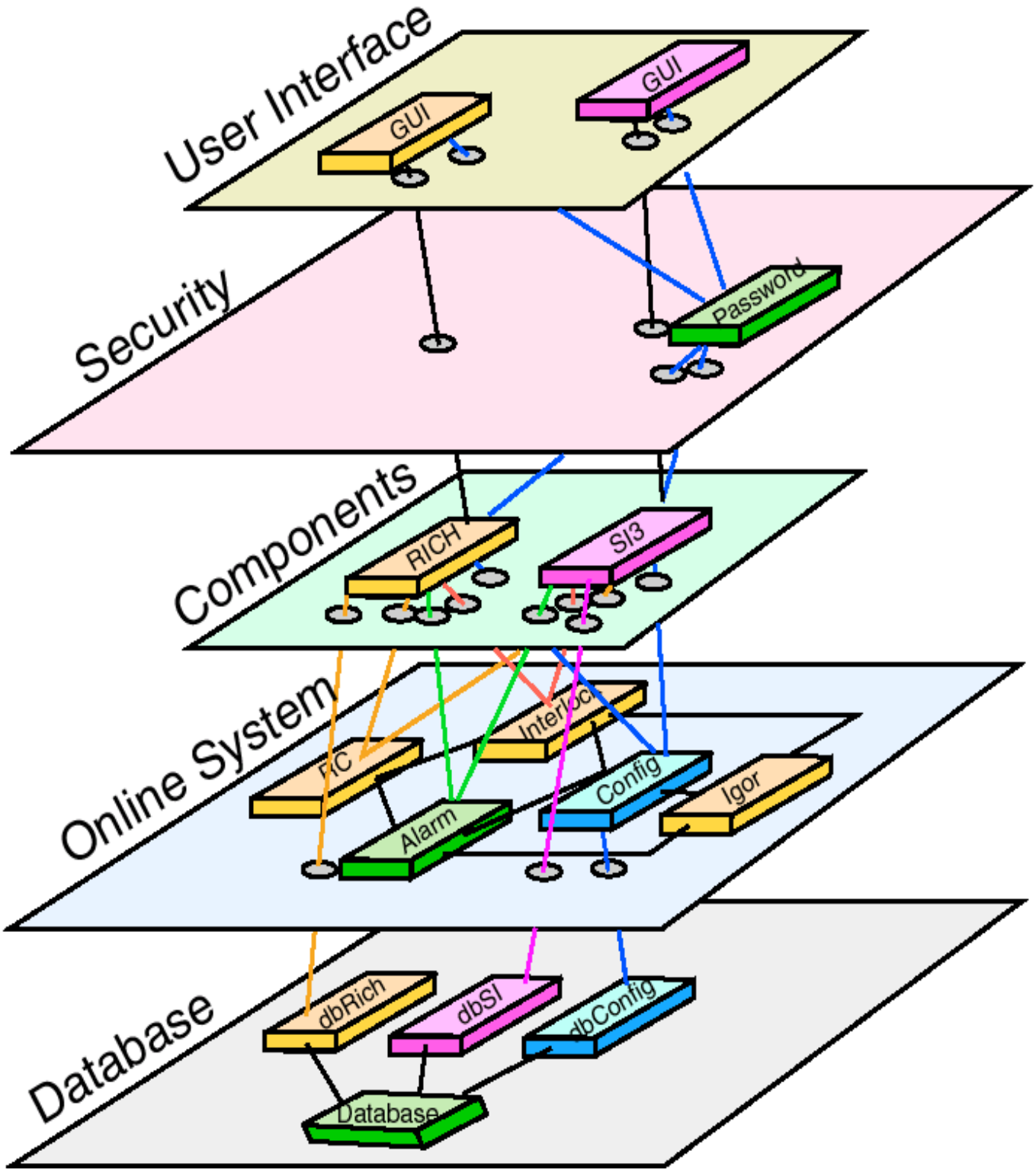
Motorola

Excellent!

Wind River (vxWorks)

well ...

CLEO III Online Software Structure



CORBA introduction from Elliots talk at CHEP98

Step 1: Hardware

```
#define ADC_CHANNEL_REG  0x8100
#define ADC_VALUE_REG    0x8104
```

Step 2: IDL

```
module VME
{
  interface ADC
   {boolean selectChannel(in long channel);
    boolean read(out long value);};
};
```

Step 3: Server

```
#include "VmeADC_s.hh"
class ADCImpl : public _sk_VME::_sk_ADC
{public:
   ADCImpl(const char *obj_name=NULL) :
       _sk_VME::_sk_ADC(obj_name)
     { }

 void selectChannel(CORBA::Long channel)
   {ADC_CHANNEL_REG = channel;}

  void read(CORBA::Long& value)
   {value = *ADC_VALUE_REG;}
}
```

Step 4: Client

```
#include "VmeADC_c.hh"
int main(int argc, char ** argv)
{// get ORB reference
  // Bind to an interface
 VME::ADC_var adc;
 adc = VME::ADC::_bind(obj_name);
 adc->selectChannel(2);
 adc->read(value);
 cout << "ADC Value " << value << endl;
 return 0;
}
```

Standard Services

Standard DAQ service are built on top of CORBA

Database

» CORBA hides implementation from user
» Objectivity 5 under Solaris (NT)

Configuration

» Load programs from DB
» Partition
» Load Constants from DB

Interlocks

» Any component can become Interlock (Source)
» Any component can react to status changes of an Interlock
» Trigger parallel actions (such as load constants) and wait for everyone to set interlock.

Alarms

» central (CORBA) alarm server hides database
» connected to Interlock system to stop data taking

Run Control

Run Statistics

» unified interface to extract (push and/or pull) information
» histograms
» dynamically view any variable, histogram in the system

Local Slow Control

Common Framework:

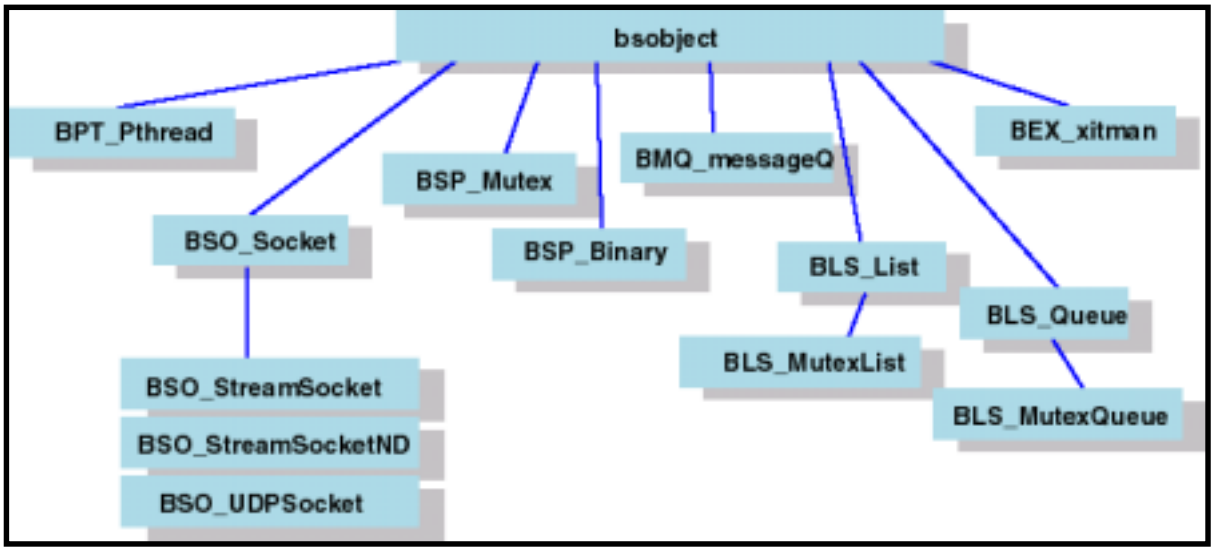
"Component Frame"

- simplifies user code development
- hides interaction with system components

Common Library:

"BASICS class library"

- hides platform dependencies



Status: complete on NT, Solaris, VxWorks

Experience with CLEO III

Code Development

C++ memory management non trivial

(important for long term stability …)

Corba services

we had to most of it ourselves

Java

JDK 2 (incl Swing) usable

Security Issues (WWW browser support)

Database:

need Objectivity expert (details hidden by CORBA)

We miss Cernlib...

Can’t afford entire suite of commercial tools

Vendor Support

Visibroker stable under NT, Solaris

We abandoned Digital Unix

Excellent VxWorks support (Highlander Inc.)

Summary

- CLEO III data acquisition is basically complete
  Data taking starts in September
- 1000 Hz trigger rate, 40 MB/s (6 MB/s) data rate,
  2% dead time
- Makes use of several novel technologies
- Databoards support new VME CBLT and MCST transfers
- Ethernet event builder is inexpensive and offers lots of computing power for added functionality
- TCP/IP provides backpressure/flow control for “free”
- FRITZ Fastbus-VME interface improves system uniformity and provides upgrade path.
- Extensive use of CORBA provides hardware independence and reduces demand for low level programming