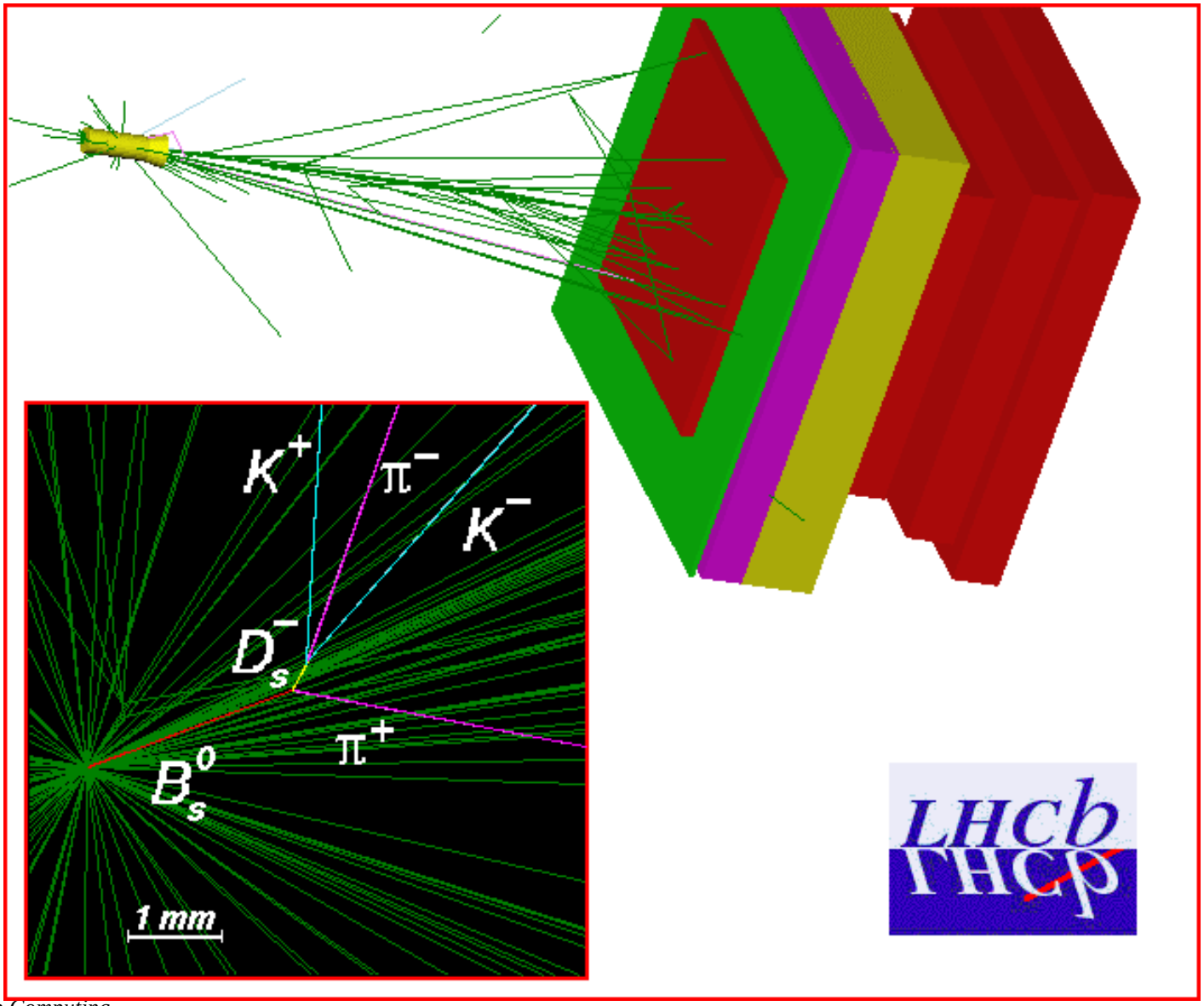
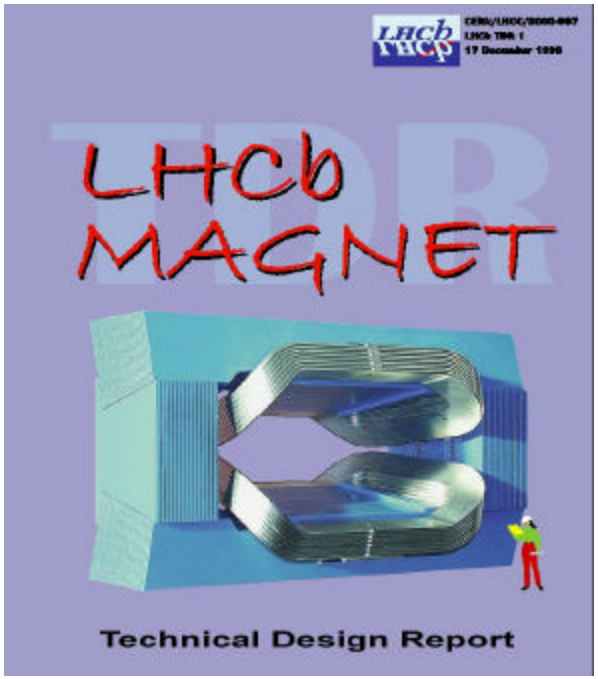

K^+
π^-
K^-
D_s^-
π^+
B_s^0
1 mm

LHCb Collaboration

France:	Clermont-Ferrand, CPPM Marseille, LAL Orsay
Germany:	Tech. Univ. Dresden, KIP Univ. Heidelberg, Phys. Inst. Univ. Heidelberg, MPI Heidelberg,
Italy:	Bologna, Cagliari, Ferrara, Firenze, Frascati, Genova, Milano, Univ. Roma I (La Sapienza), Univ. Roma II(Tor Vergata)
Netherlands:	NIKHEF
Poland:	Cracow Inst. Nucl. Phys., Warsaw Univ.
Spain:	Univ. Barcelona, Univ. Santiago de Compostela
Switzerland:	Univ. Lausanne, Univ. Zürich
UK:	Univ. Bristol, Univ. Cambridge, Univ. Edinburgh, Univ. Glasgow, IC London, Univ. Liverpool, Univ. Oxford, RAL
CERN	
Brazil:	UFRJ
China:	IHEP (Beijing), Tsinghua Univ. (Beijing)
Romania:	IFIN-HH Bucharest
Russia:	BINR (Novosibirsk), INR, ITEP,Lebedev Inst., IHEP,PNPI(Gatchina)
Ukraine:	Inst. Phys. Tech. (Kharkov), Inst. Nucl. Research (Kiev)

Requirements on Data Rates and Computing Capacities

LHCb Technical Design Reports



Submitted:
January 2000
Recommended by LHCC:
March 2000
Approved by RB:
April 2000

Submitted:
September 2000
Recommended:
November 2000

Submitted:
September 2000
Recommended:
November 2000

Defining the architecture

- Issues to take into account
  - Object persistency
  - User interaction
  - Data visualization
  - Computation
  - Scheduling
  - Run-time type information
  - Plug-and-play facilities
  - Networking
  - Security

Architectural Styles

- General categorization of systems [2]

user-centric	focus on the **direct visualization and manipulation** of the objects that define a certain domain
data-centric	focus upon preserving the **integrity of the persistent objects** in a system
computation-centric	focus is on the **transformation of objects** that are interesting to the system

Our applications have elements of all three. Which one dominates?

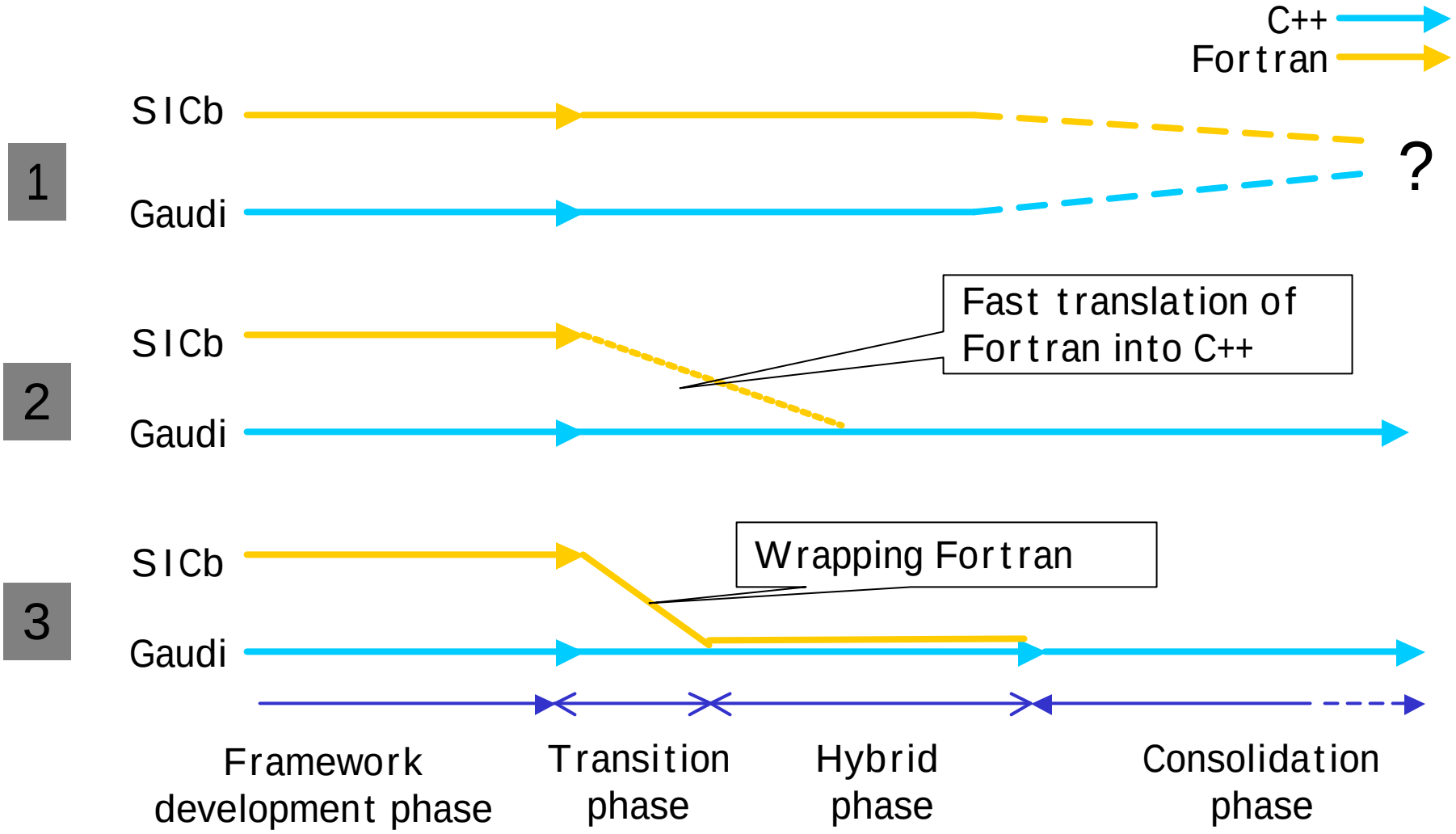
Getting Started

- First crucial step was to appoint an architect - ideally skills as:
  - OO mentor, domain specialist, leadership, visionary
- Started with small design team ~ 6 people, including :
  - developers , librarian, use case analyst
- Control activities through visibility and self discipline
  - meet regularly - in the beginning every day, now once per week
- Collect URs and scenarios, use to validate the design
- Establish the basic design criteria for the overall architecture
  - architectural style, flow of control, specification of interfaces

Development Process

- Incremental approach to development
  - new release every few (~ 4) months
  - software workshop timed to coincide with new release
- Development cycle is user-driven
  - Users define priority of what goes in the next release
  - Ideally they use what is produced and give rapid feedback
  - Frameworks must do a lot and be easy to use
- Strategic decisions taken following thorough review (~1 /year)
- Releases accompanied by complete documentation
  - presentations, tutorials
  - URD, reference documents, user guides, examples

Possible migration strategies



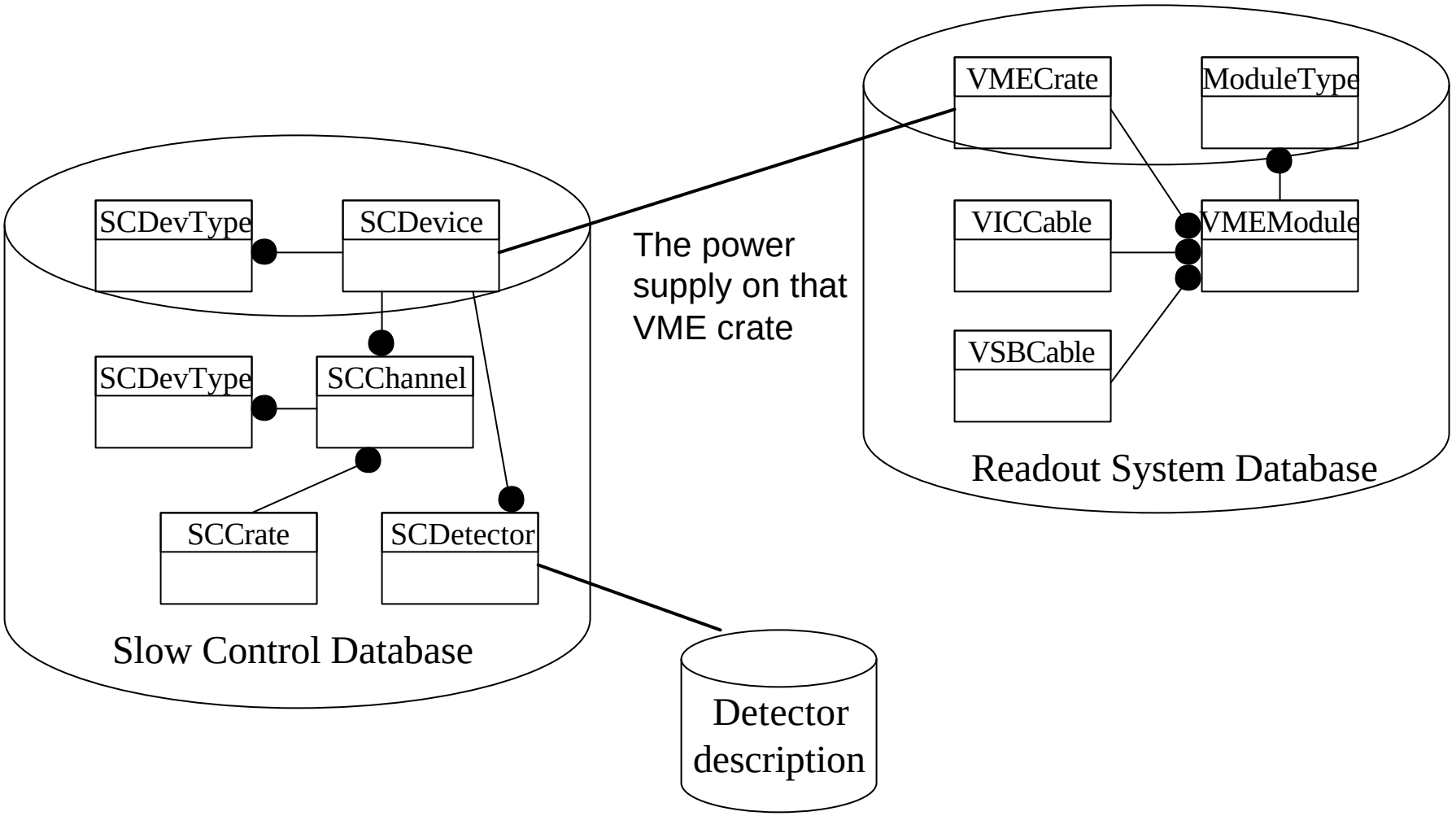
How to proceed?

- Physics Goal:
  - To be able to run new tracking pattern recognition algorithms written in C++ in production with standard FORTRAN algorithms in time to produce useful results for the RICH TDR.
- Software Goal
  - To allow software developers to become familiar with GAUDI and to encourage the development of new software algorithms in C++.
- Approach
  - choose strategy 3
  - start with migration of reconstruction and analysis code
  - simulation will follow later

New Reconstruction Program - BRUNEL

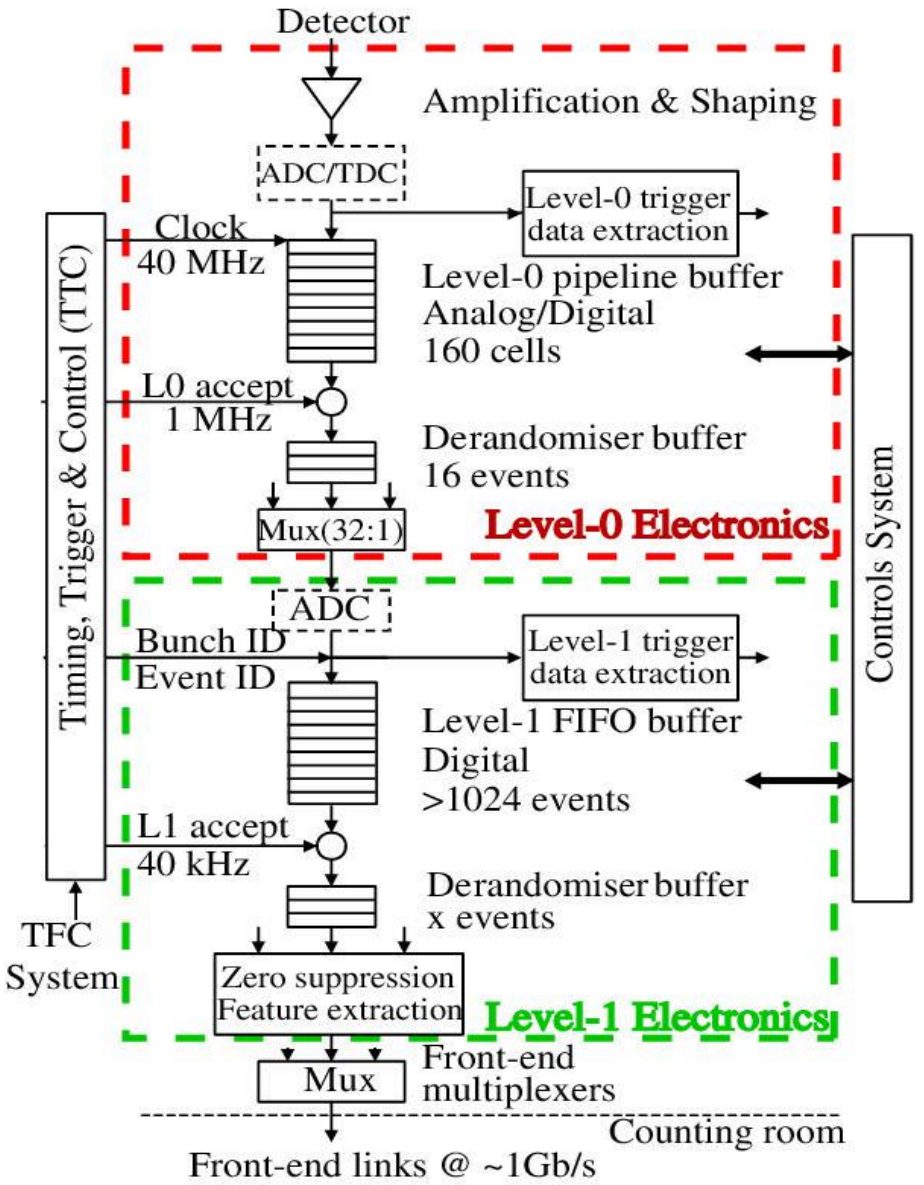
- Benefits of the approach
- A unified development and production environment
  - As soon as C++ algorithms are proven to do the right thing, they can be brought into production in the official reconstruction program
- Early exposure of all developers to Gaudi framework
- Increasing functionality of OO 'DST'
  - As more and more of the event data become available in Gaudi, it will become more and more attractive to perform analysis with Gaudi
- A smooth transition to a C++ only reconstruction

Integrated System - databases



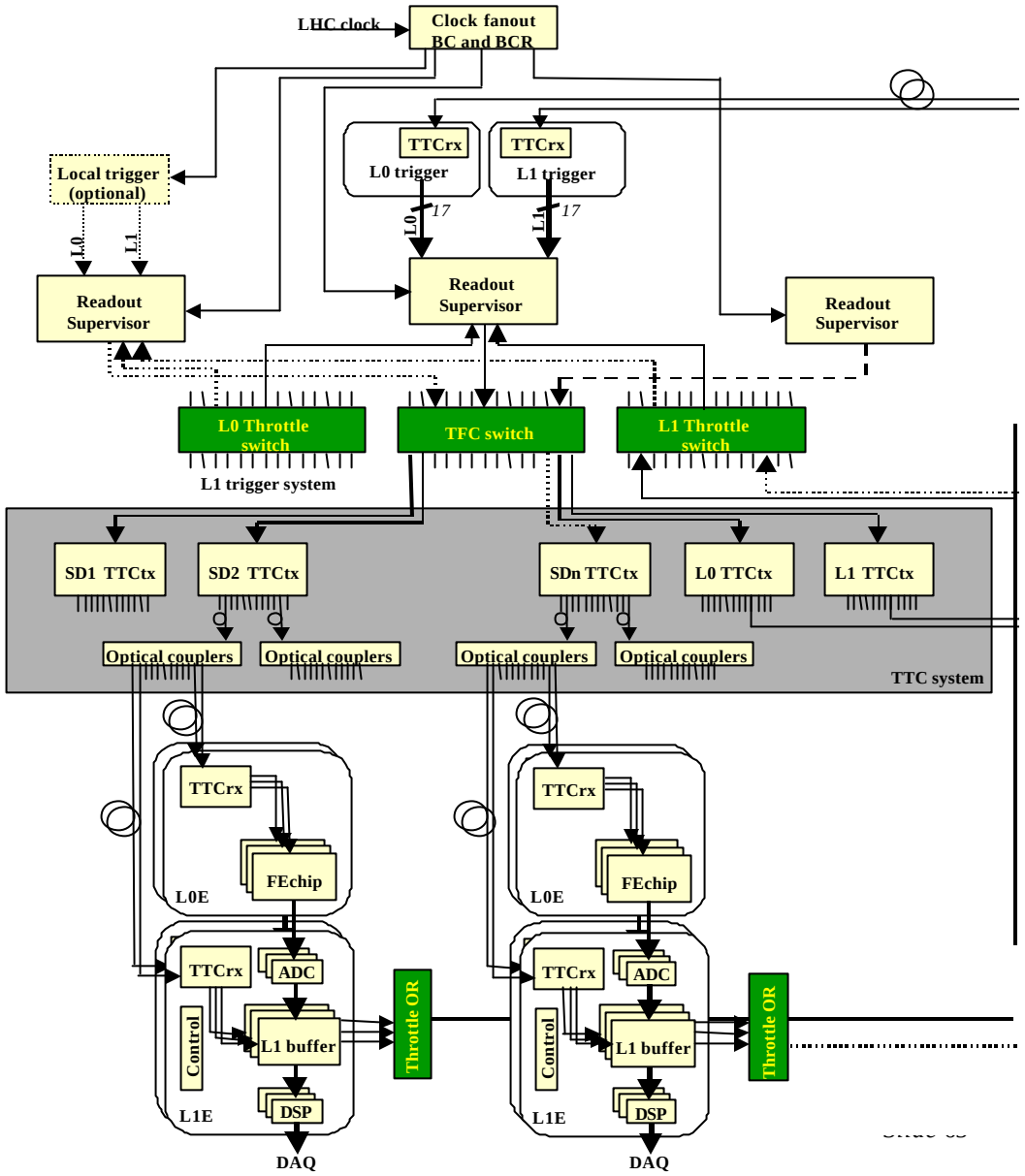
Frontend Electronics

- Data Buffering for Level-0 latency
- Data Buffering for Level-1 latency
- Digitization and Zero Suppression
- Front-end Multiplexing onto Front-end links
- Push of data to next higher stage of the readout (DAQ)



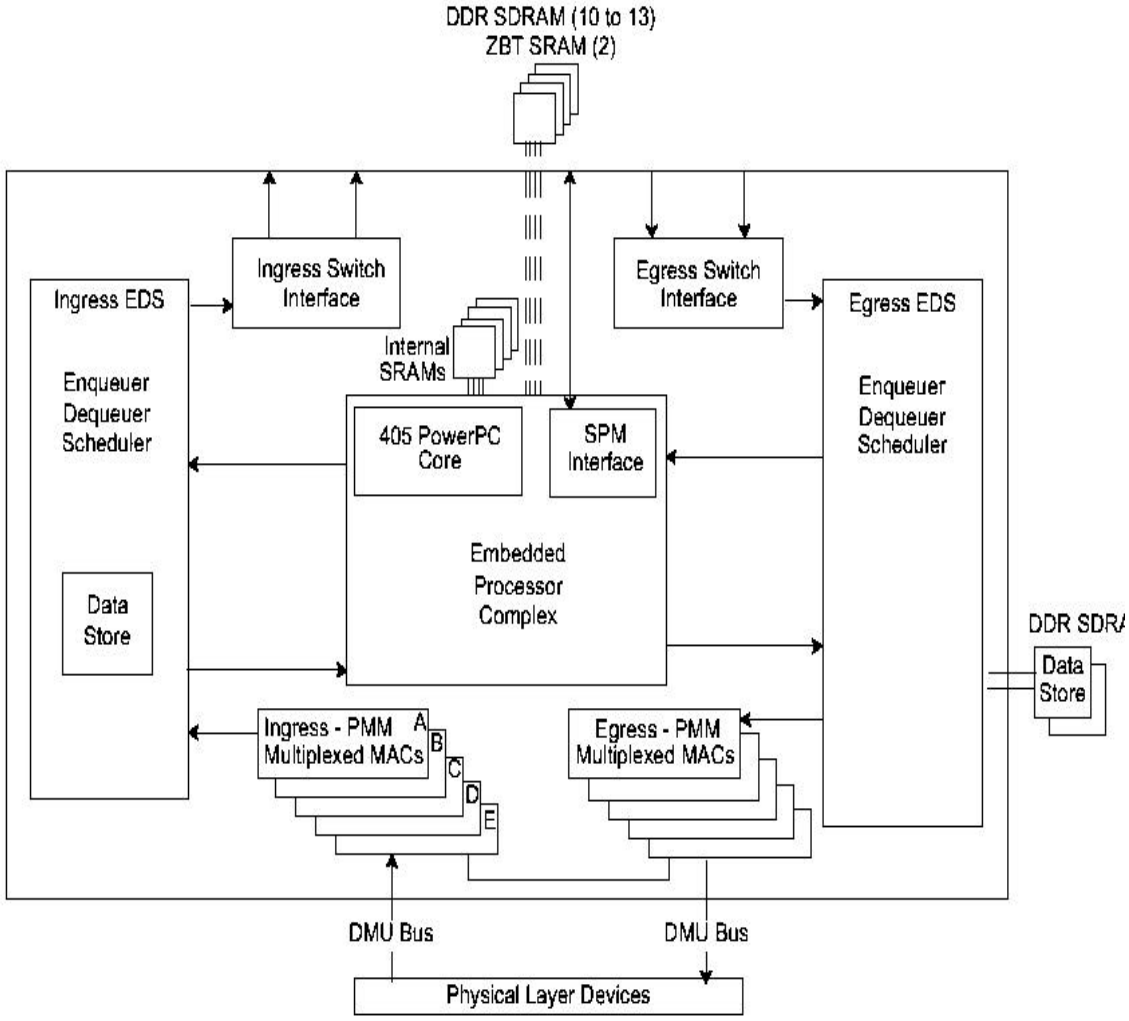
Timing and Fast Control

- Provide common and synchronous clock to all components needing it
- Provide Level-0 and Level-1 trigger decisions
- Provide commands synchronous in all components (Resets)
- Provide Trigger hold-off capabilities in case buffers are getting full
- Provide support for partitioning (Switches, ORs)



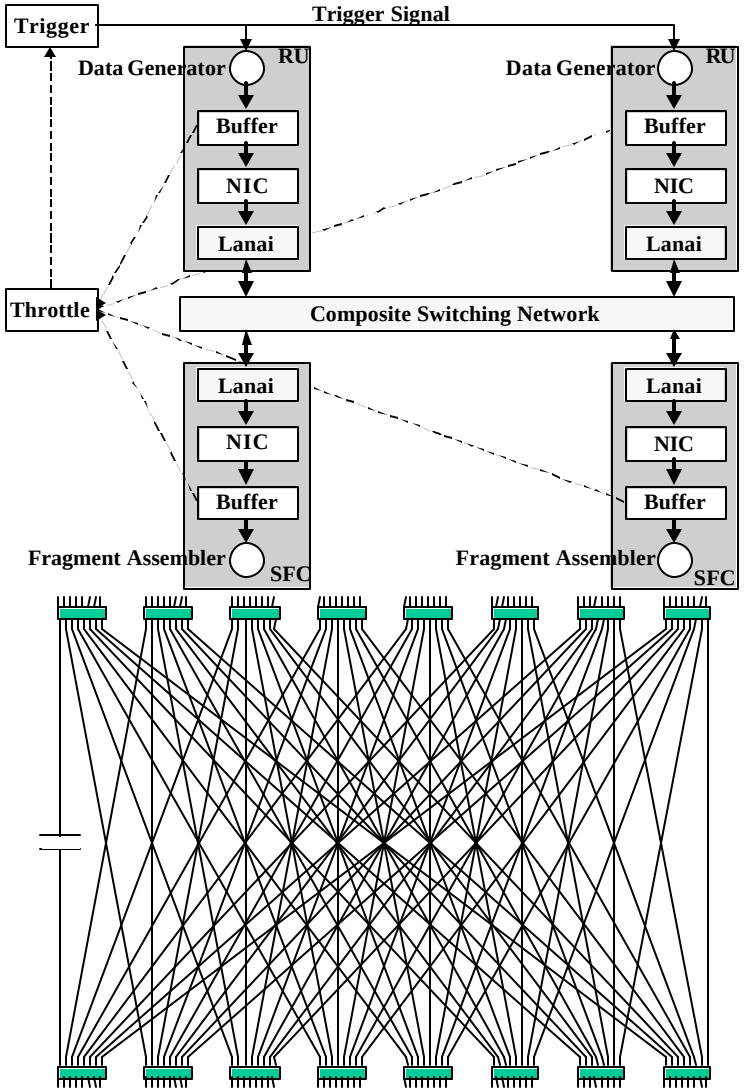
IBM NP4GS3

- Features
  - 4 x 1Gb full duplex Ethernet MACs
  - 16 special purpose RISC processors @ 133 MHz with 2 hw threads each
  - 4 processor (8 threads) share 3 co-processors for special functions
    - Tree search
    - Memory move
    - Etc.
  - Integrated 133 MHz Power PC processor
  - Up-to 64 MB external RAM



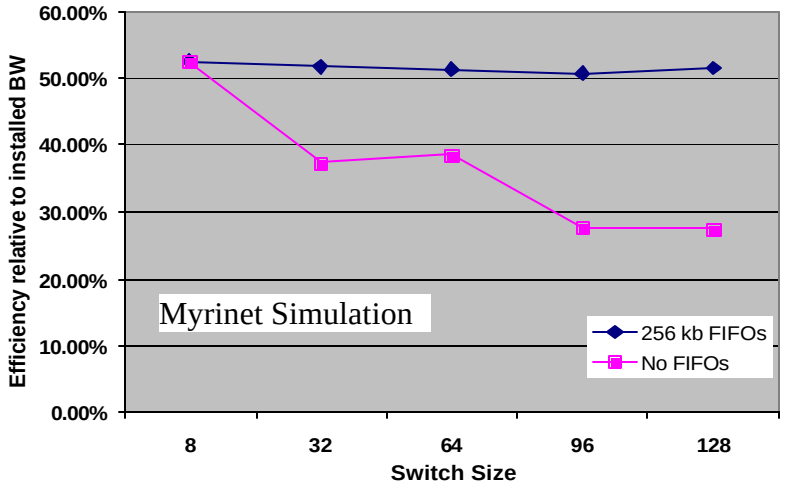
Event Building Network Simulation

- Simulated technology: Myrinet
  - Nominal 1.28 Gb/s
  - Xon/Xoff flow control
  - Switches:
    - ideal cross-bar
    - 8x8 maximum size (currently)
    - wormhole routing
    - source routing
    - No buffering inside switches
- Software used: Ptolemy discrete event framework
- Realistic traffic patterns
  - variable event sizes
  - event building traffic

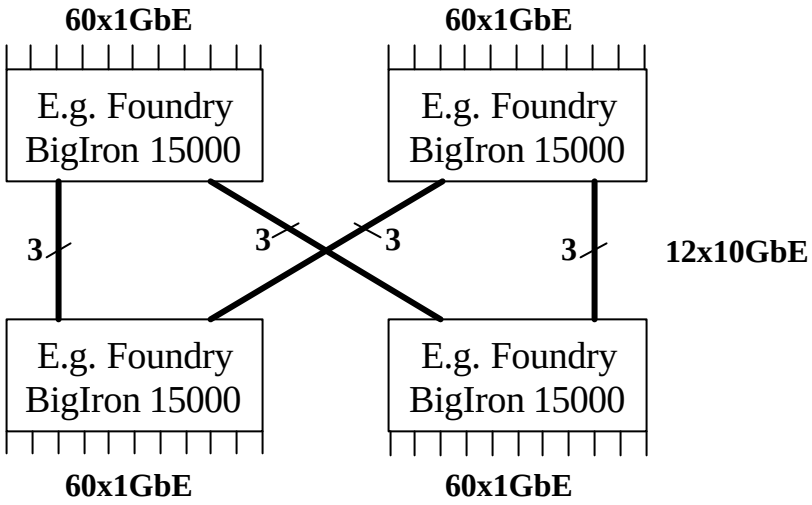


Event Building Activities

- Studied Myrinet
  - Tested NIC event-building
  - simulated switching fabric of the size suitable for LHCb
    Results show that switching network could be implemented (provided buffers are added between levels of switches)



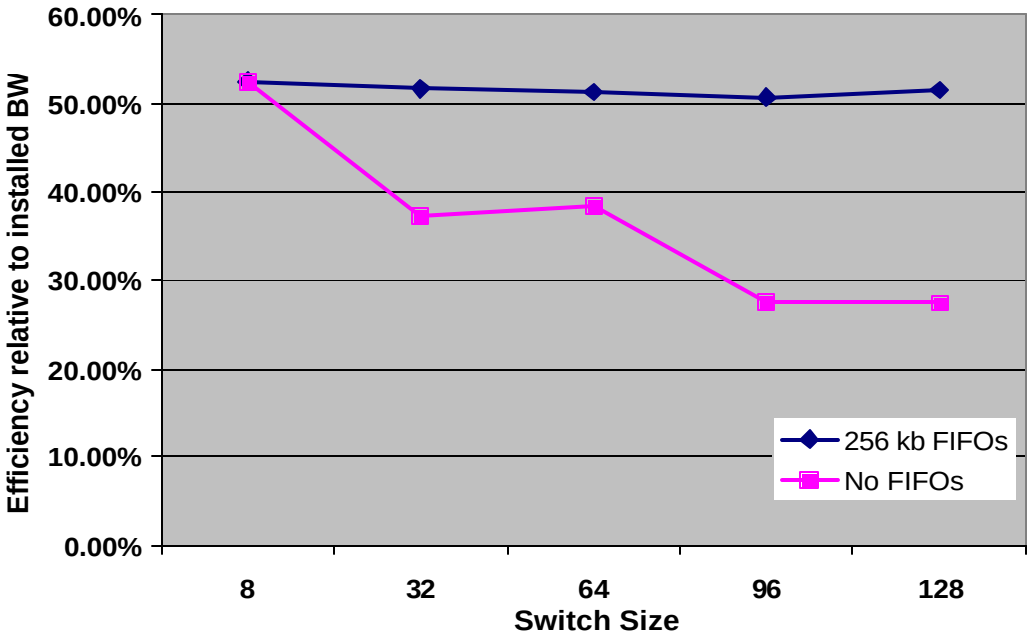
- Currently focussing on xGb Ethernet
  - Studying smart NICs (-> Niko's talk)
  - Possible switch configuration for LHCb with ~today's technology
    (to be simulated...)



Multiple Paths between sources and destinations!

Network Simulation Results

Results don't depend strongly on specific technology (Myrinet), but rather on characteristics (flow control, buffering, internal speed, etc)

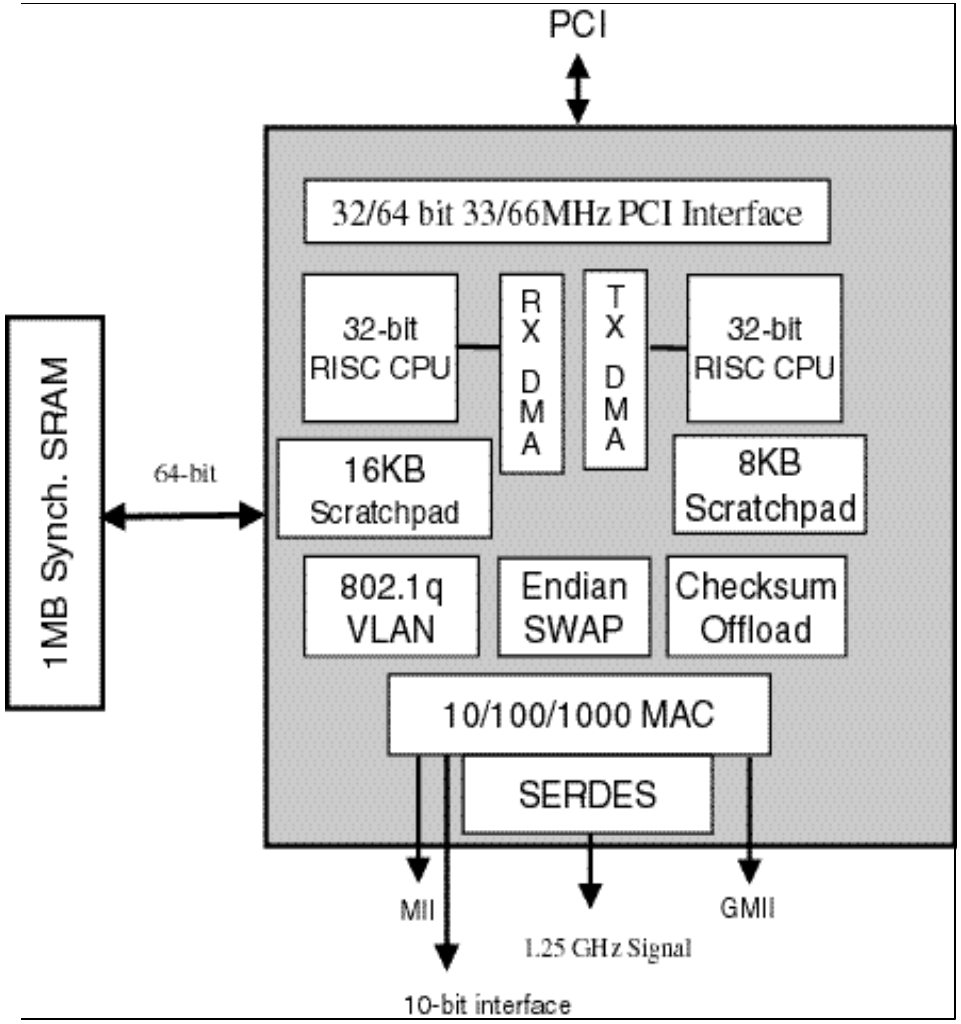


Switch Size	Fifo Size	Switching Levels	Efficiency
8x8	NA	1	52.5%
32x32	0	2	37.3%
32x32	256 kB	2	51.8%
64x64	0	2	38.5%
64x64	256 kB	2	51.4%
96x96	0	3	27.6%
96x96	256 kB	3	50.7%
128x128	0	3	27.5%
128x128	256 kB	3	51.5%

FIFO buffers between switching levels allow to recover scalability

50 % efficiency "Law of nature" for these characteristics

Alteon Tigon 2

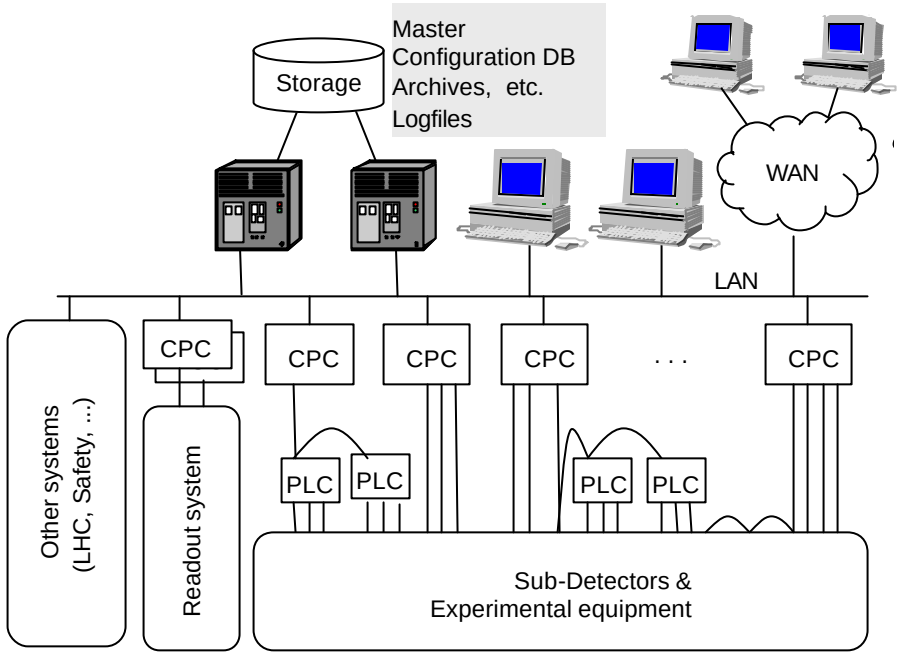


- Features
  - Dual R4000-class processor running at 88 MHz
  - Up to 2 MB memory
  - GigE MAC+link-level interface
  - PCI interface
- Development environment
  - GNU C cross compiler with few special features to support the hardware
  - Source-level remote debugger

Controls System

Common integrated controls system

- Detector controls
  - High voltage
  - Low voltage
  - Crates
  - Alarm generation and handling
  - etc.
- DAQ controls
  - RUN control
  - Setup and configuration of all components (FE, Trigger, DAQ, CPU Farm, Trigger algorithms,...)
  - Consequent and rigorous separation of controls and DAQ path



Same system for both functions!

Scale: ~100-200 Control PCs
many 100s of Credit-Card PCs

By itself sizeable Network!
Most likely Ethernet