The High Level Filter of the H1 Experiment at HERA

thanks to
R.Gerhards,C.Grab,S.Levonian,J.Martyniak,T.Mkrtchyan,C.Nowak,J.Nowak,
P.Fleischmann,M.Vorobiev

Alan Campbell CHEP'04

Outline

H1 Experiment

L45 trigger

New L45 scheme

Event Repository

Data Flow

Process and Corba Object Management

Monitoring - Online Histograms, Event display, Emergency Messages, Output Log

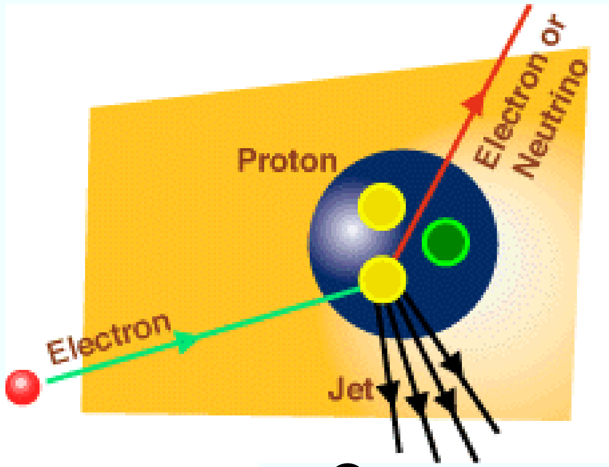
Calibration

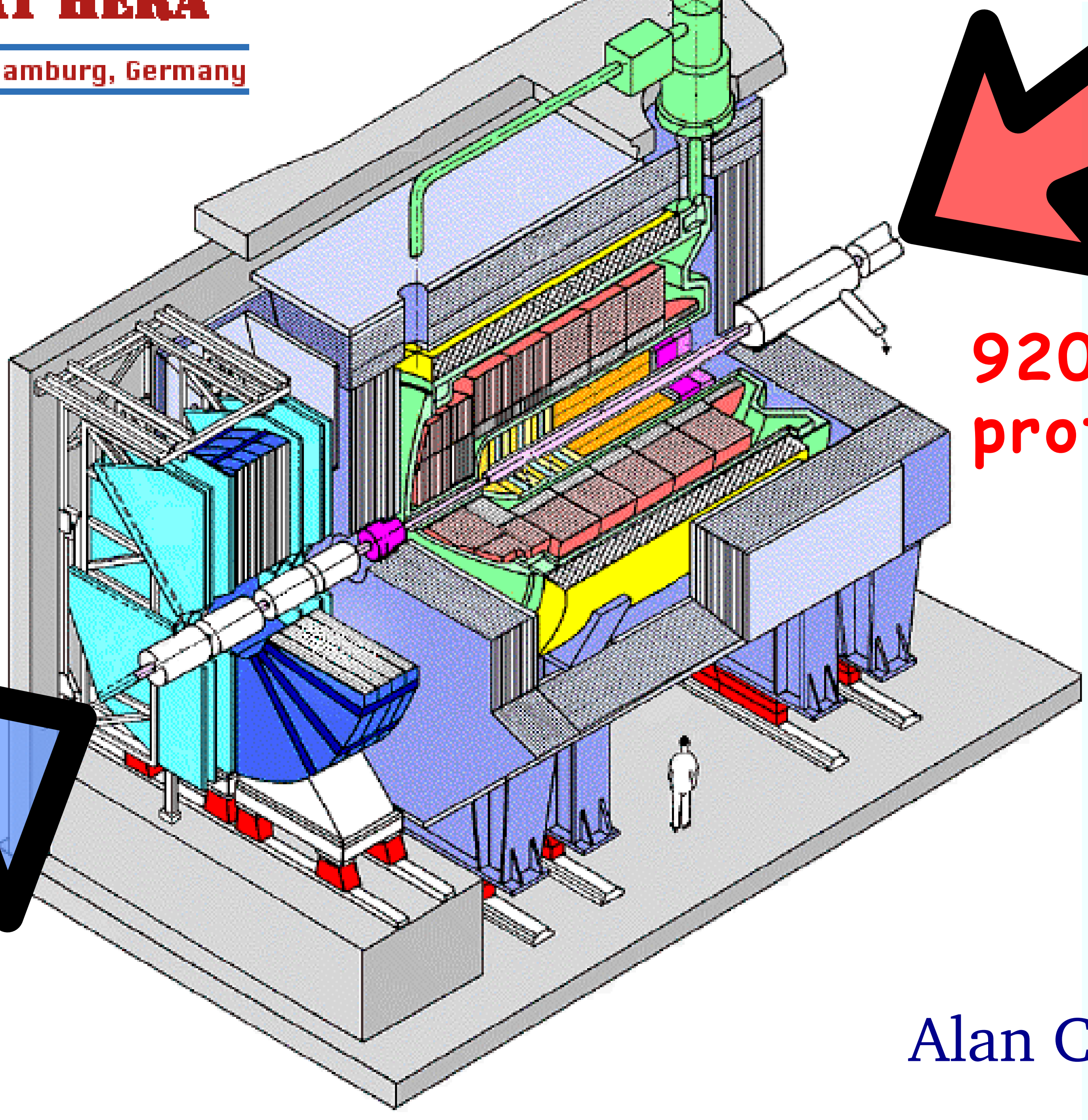
Trigger Algorithm Steering

Conclusions

THE H1 EXPERIMENT AT HERA

H1 Collaboration at DESY, Notkestr.85, D-22607 Hamburg, Germany



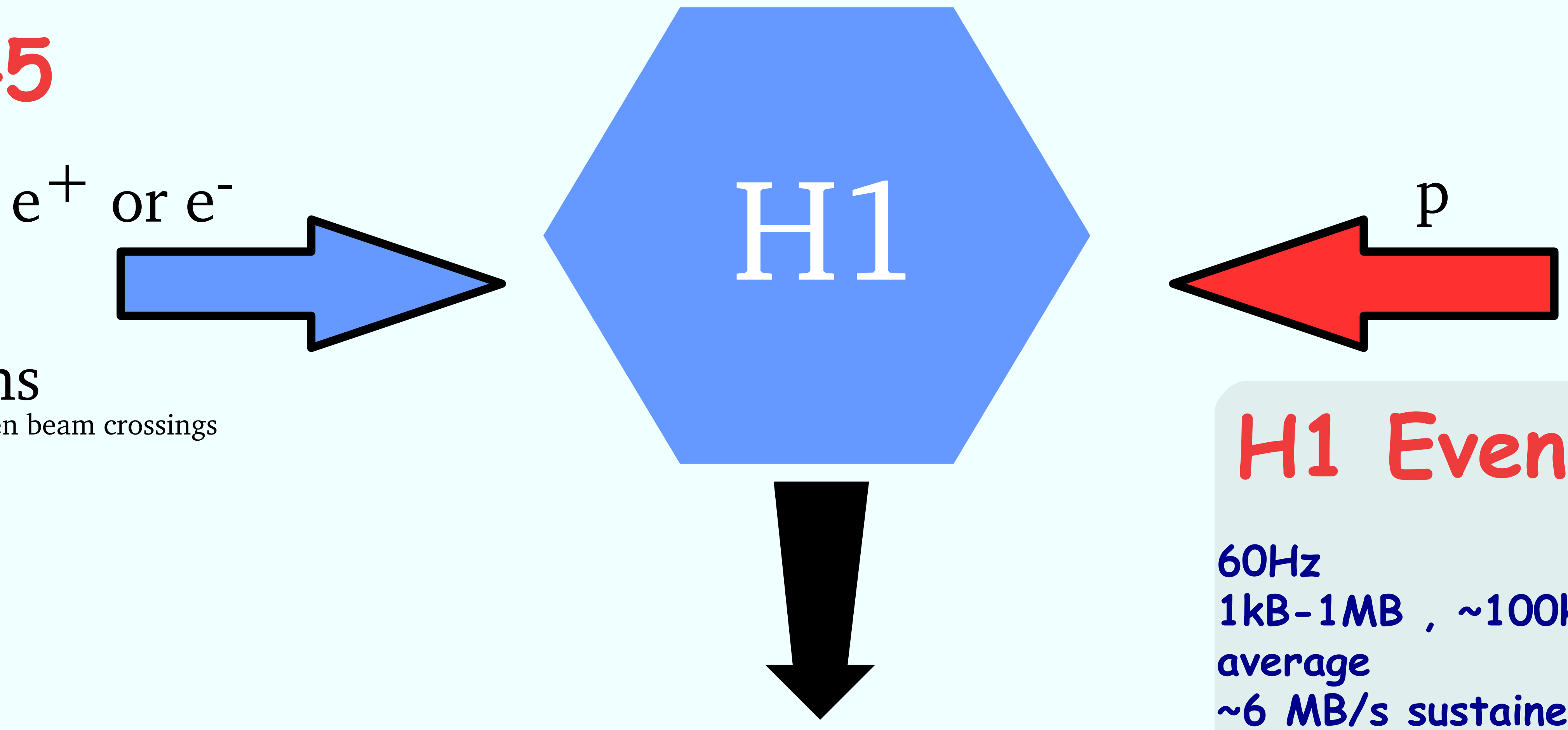


920 GeV proton

30 GeV positron or electron

Alan Campbell

L45



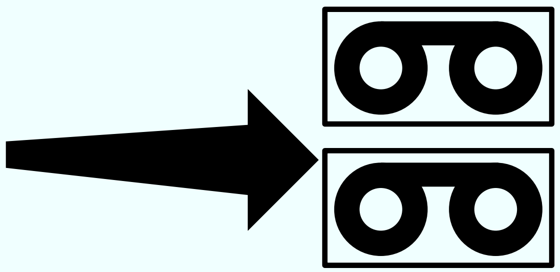
96ns
between beam crossings

H1 Events

60Hz
1kB-1MB , ~100kB average
~6 MB/s sustained average

L45 trigger

Full event reconstruction
Background rejection (eg vertex position)
Event Classification (assign to output stream)
Finders - select special physics channels
Downscale soft physics
Monitor & Calibration



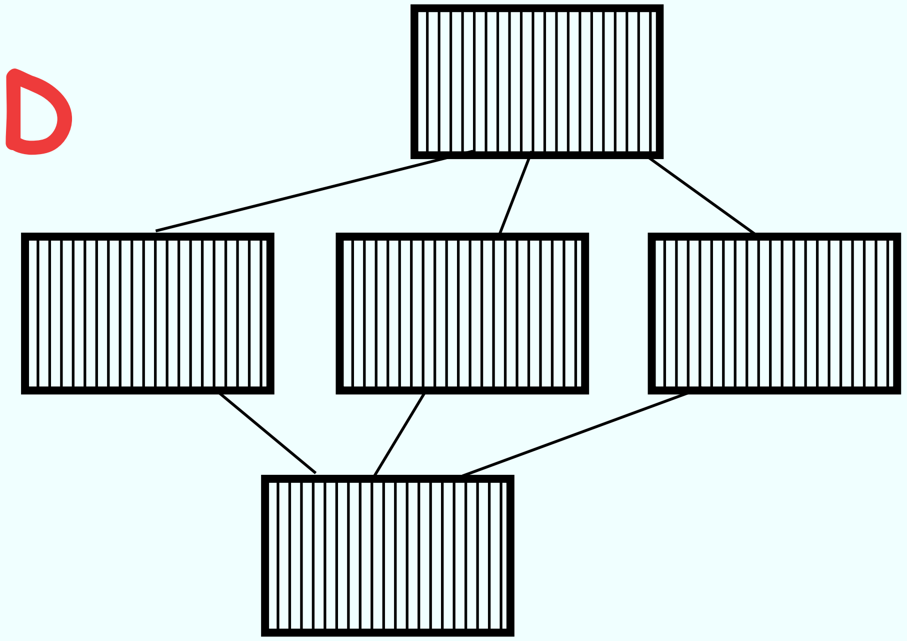
Storage

~3 MB/s
RAW
DST
RANDOM
LED...

Data taking in "runs" of few minutes to one hour.
A run has fixed trigger and readout settings.

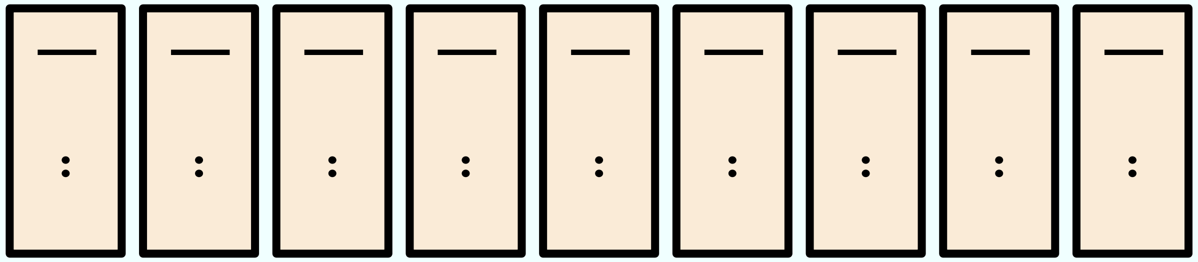
L45 upgrade HERAII

OLD



VMEbus processors
standalone programs

NEW



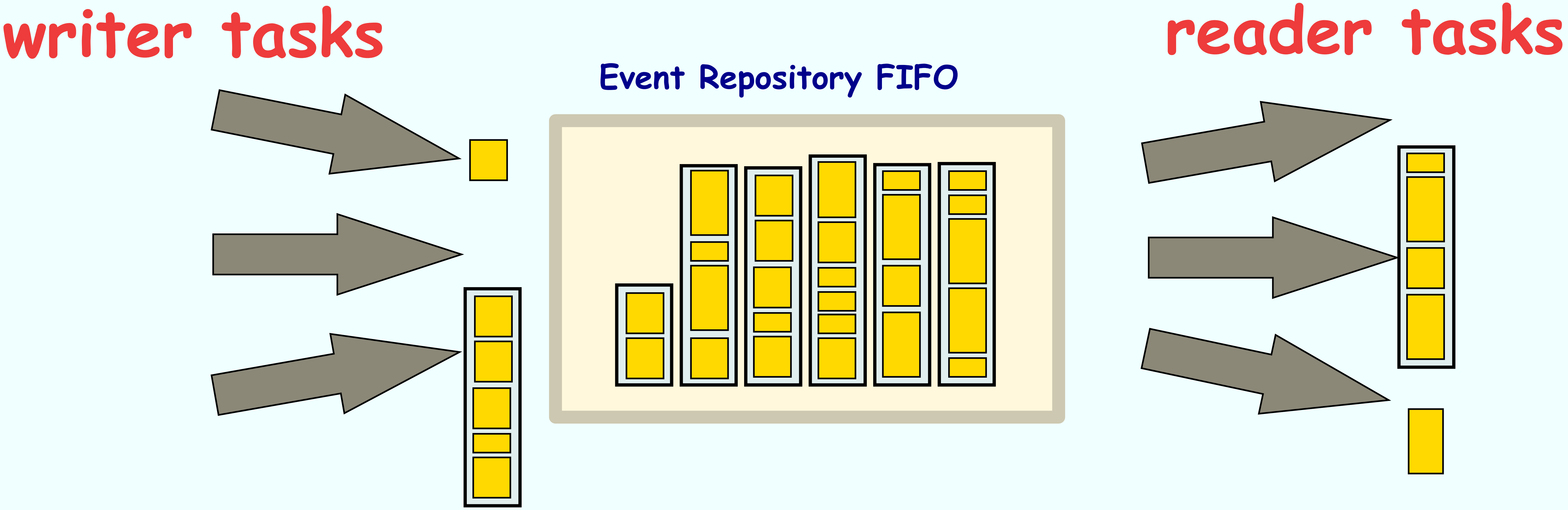
Networked PCs

-Same platform as offline computing (identical software) -resources can be shared
-lower cost -expandable network & cpu -same framework for reprocessing

Approach: standards based event distribution framework
CORBA for data and control -> multiple language bindings
C++/Fortran/C -> link binary code from H1 standard libraries
Python -> control & setup scripts
Java -> histogram collection & display, run control

Investigations showed that data transfer via CORBA is fast (>7MB/s on 100Mbit network) with <1ms/call overhead => transfer in chunks of ~500kB.

Event Repository - 1.Basics

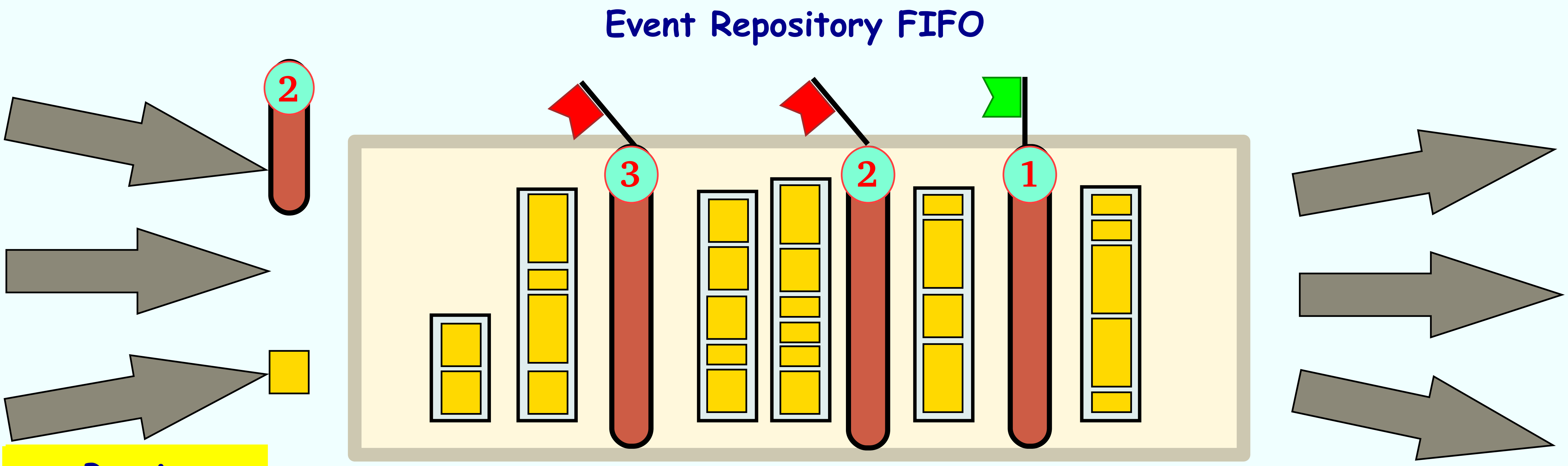


write_event
write_sequence

read_event
read_sequence

a first-in first-out event store
read and write similar to sequential file access
events stored in repository as suitably sized sequences
sequences are created when single events are inserted
only entire sequences are transferred between repositories via network
multiple writers & readers simultaneously - multi-threaded orb
no expensive data copy required
used both for event collection and distribution

Event Repository - 2.Synchronisation

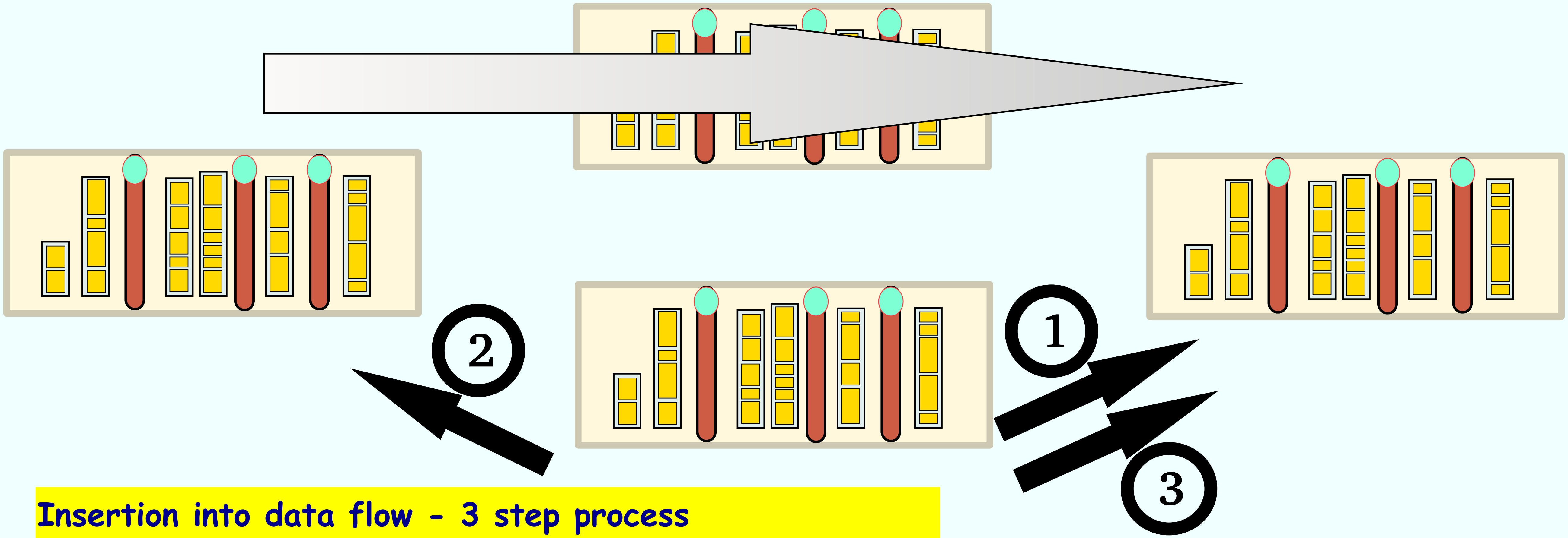


Barriers

seperate event flow into blocks (eg runs), events stay within a block
all output events in a block arrive before any event from the next block
events may be dropped from (eg trigger reject) or added to (eg extra calibration data) a block
barriers are eg run start/stop, file begin/end, nth event calibration trigger

Barriers assigned incremental number when first written to its source repository
all writers must write a barrier before any readers can read it
writers write event sequences in front of a barrier which they have not yet written
as soon as all writers have written a barrier it becomes readable
barriers are not removed until read by all readers
a reader can read sequences behind barriers he has already read
each barrier is distributed to all repositories
repositories must be large and barriers infrequent to not hinder data flow

Event Repository - 3.Linking



Insertion into data flow - 3 step process

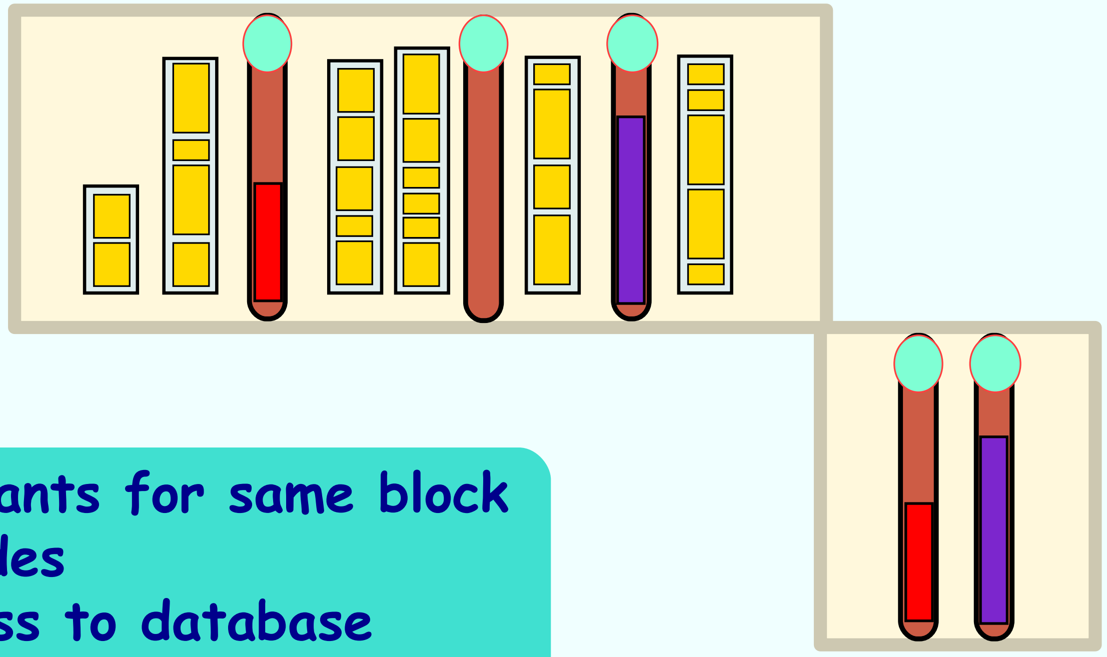
1. tell sink repository we will deliver front-most non-readable barrier
2. tell source repository we will read from front-most barrier
3. tell sink we will deliver only from barrier number given by source repository

Similar care needed to detach from flow normally or abruptly

Event Repository - 4.Barriers with Data

Barriers are "broadcast" to all repositories and mark timestamp in dataflow.
Data Attached to barriers provides distribution of constants.

Event Repository FIFO

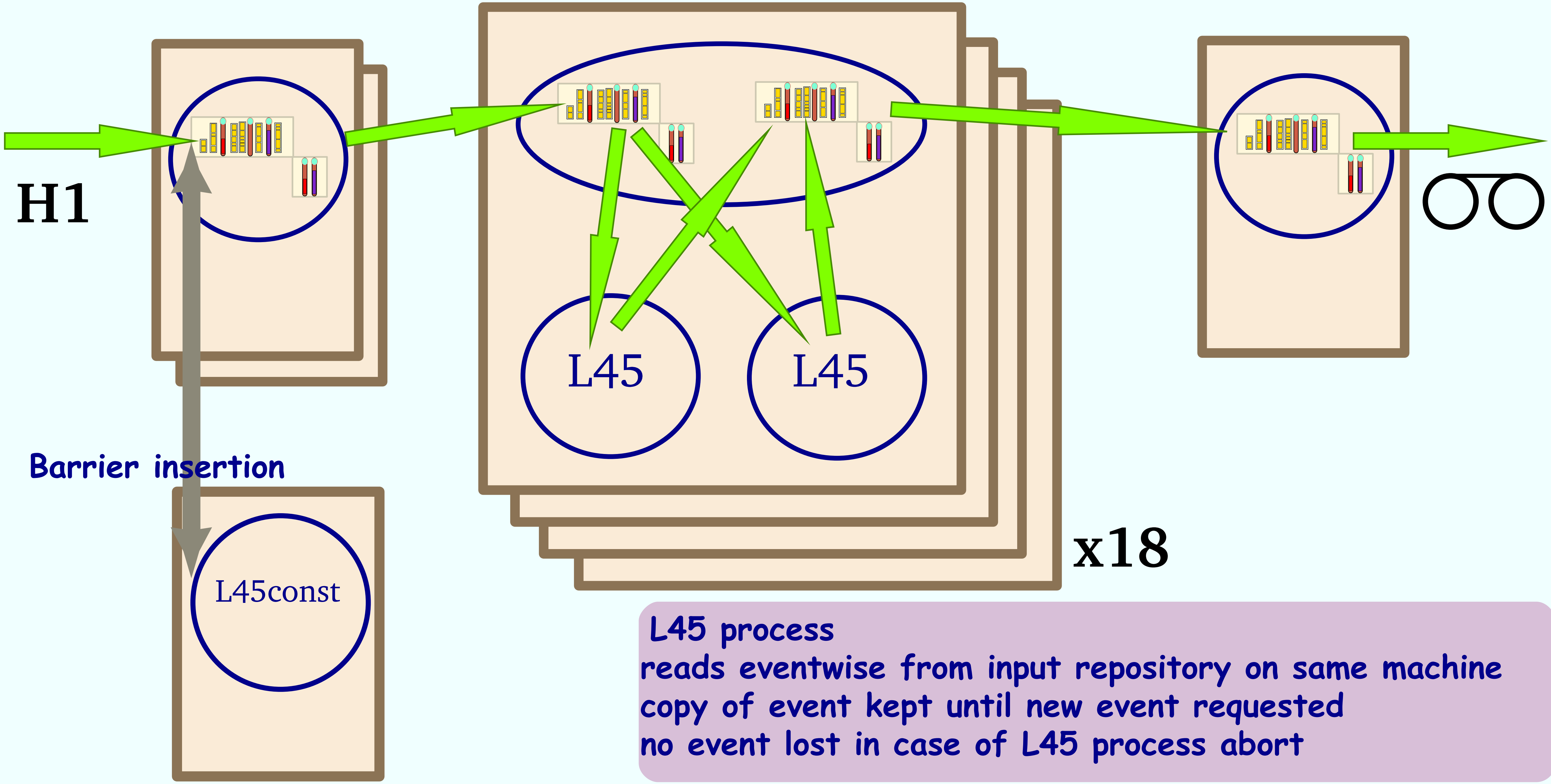


Persistent Barrier Cache

ensures same constants for same block on all processing nodes
avoids multiple access to database
timely distribution of run-start records ie run settings

When barrier with data is removed from repository it enters a cache, replacing last barrier of this type.
New readers first read the barriers in the cache to obtain all current constants.

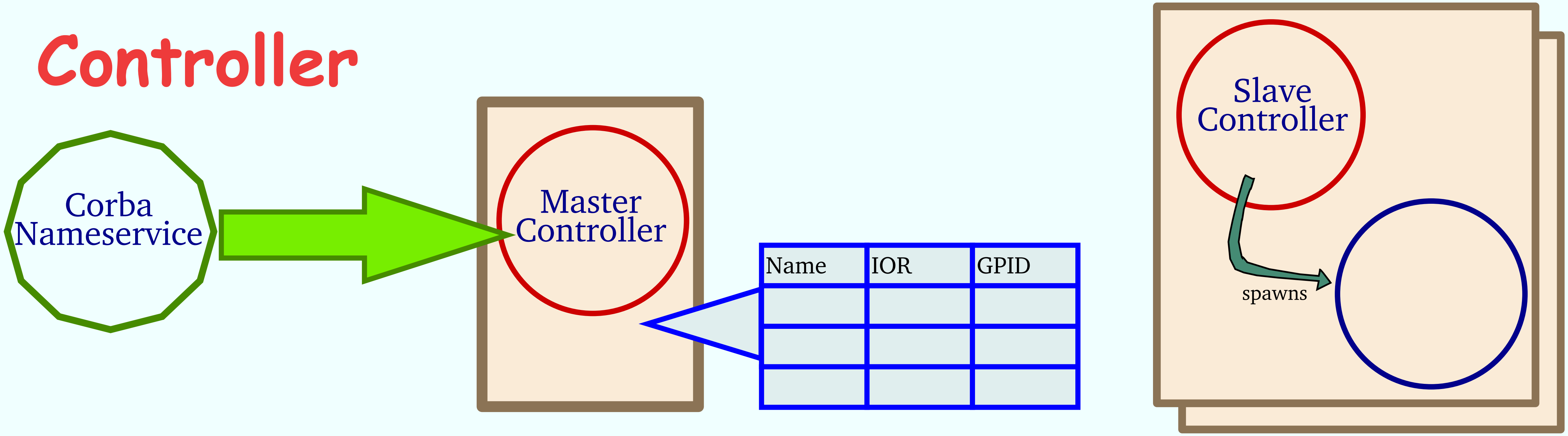
Overall Dataflow



L45 process
reads eventwise from input repository on same machine
copy of event kept until new event requested
no event lost in case of L45 process abort

l45const - extra barrier creation process
allows for multiple input processes
inserts additional barriers in data flow if new constants available

Controller



Name	IOR	GPID

Controller - process creation and corba object handling (python & omnipy)

Master controller

starts slave controllers via ssh

provides global process ids

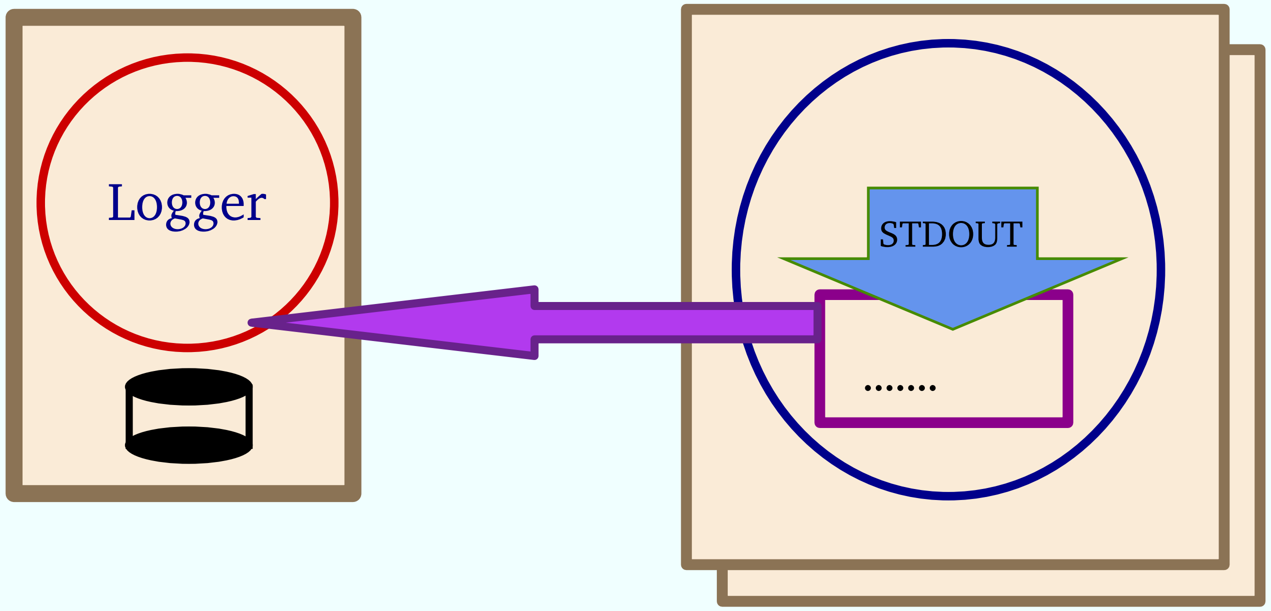
maintains lists of object references eg of event repositories, histogram servers

maintains list of repositories and their reader/writers

Slave controllers start processes for master, check & restart processes

controllers are restartable

Logger



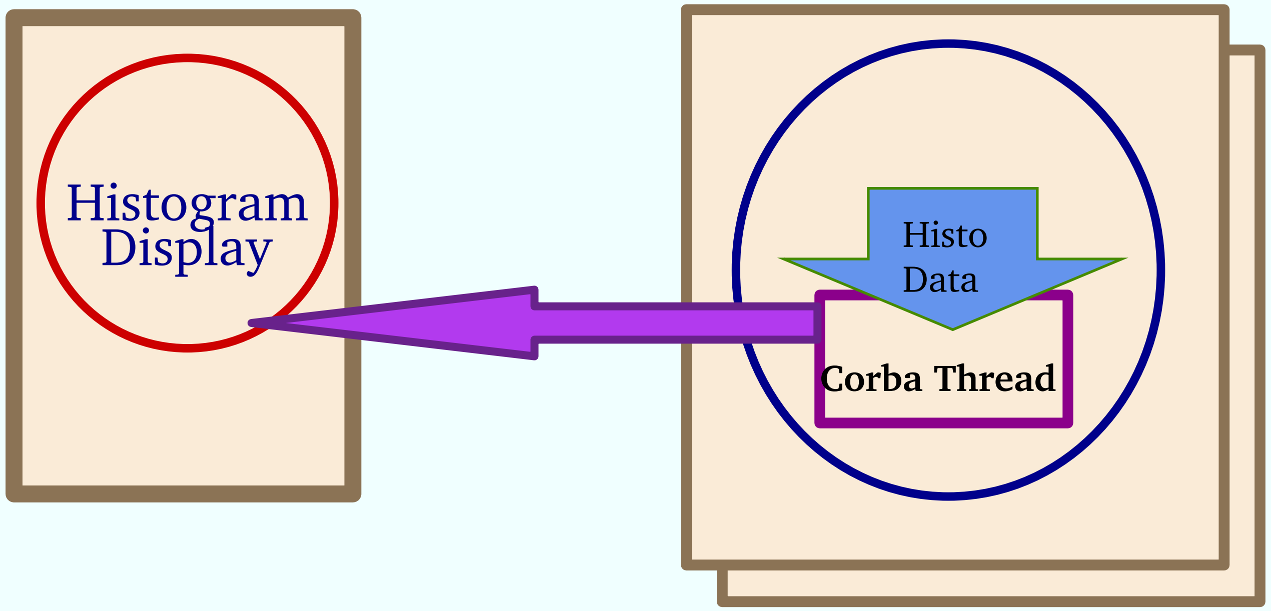
C++ thread reads STDOUT

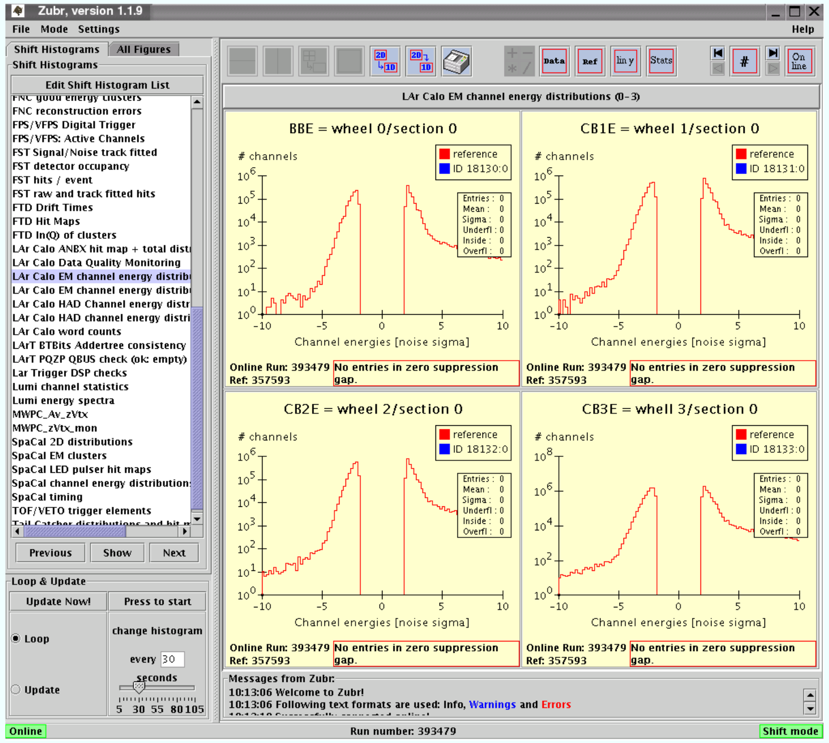
collects lines

transfers to logger via corba

logger process dumps to disk

Online Histograms





Shift Histograms

- FNC good energy clusters
- FNC reconstruction errors
- FPS/VFPS Digital Trigger
- FPS/VFPS: Active Channels
- FST Signal/Noise track fitted
- FST detector occupancy
- FST hits / event
- FST raw and track fitted hits
- FTD Drift Times
- FTD Hit Maps
- FTD ln(Q) of clusters
- LAr Calo ANBX hit map + total distr
- LAr Calo Data Quality Monitoring
- LAr Calo EM channel energy distribu
- LAr Calo EM channel energy distribu
- LAr Calo HAD Channel energy distr
- LAr Calo HAD channel energy distri
- LAr Calo word counts
- LArT BTBits Addertree consistency
- LArT PQZP QBUS check (ok: empty)
- Lar Trigger DSP checks
- Lumi channel statistics
- Lumi energy spectra
- MWPC_Av_zVtx
- MWPC_zVtx_mon
- SpaCal 2D distributions
- SpaCal EM clusters
- SpaCal LED pulser hit maps
- SpaCal channel energy distribution
- SpaCal timing
- TOF/VETO trigger elements
- Tail Catcher distributions and hit m

Online Run: 393479 Ref: 357593 No entries in zero suppression gap.

Online Run: 393479 Ref: 357593 No entries in zero suppression gap.

Online Run: 393479 Ref: 357593 No entries in zero suppression gap.

Online Run: 393479 Ref: 357593 No entries in zero suppression gap.

Messages from Zubr:

```
10:13:06 Welcome to Zubr!
10:13:06 Following text formats are used: Info, Warnings and Errors
```

Run number: 393479

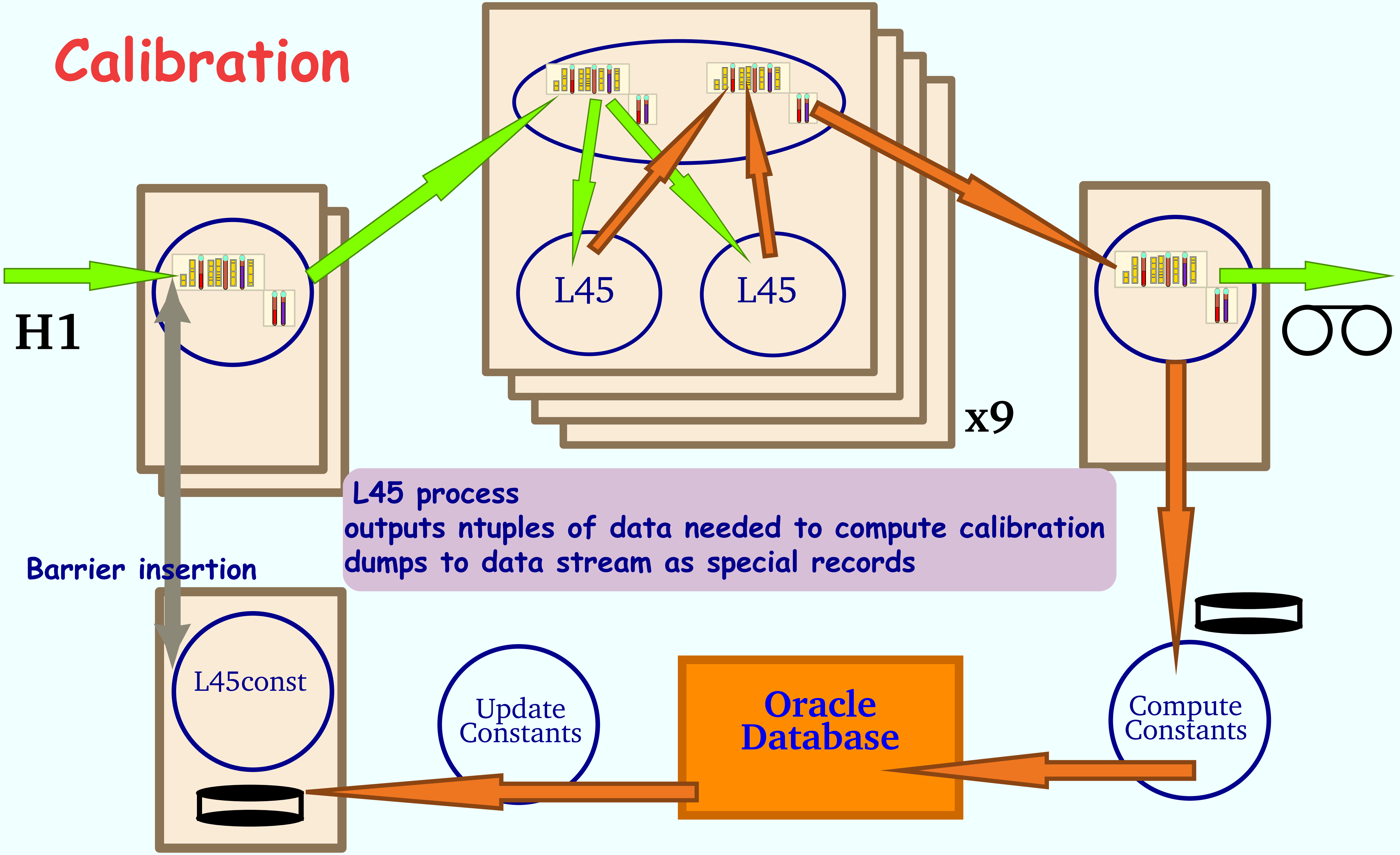
Corba thread C++ calls fortran histogram package (LOOK)
LOOK modified to add recursive mutex
Histogram display fetches and sums data from all processes(Java, JAS, corba)
New: web access via corba web server in python (omnipy,biggles,svg)

Online Event Display

rare events may be selected for the online event display
selection criteria are sent as 'constants' via l45const
events selected in L45 are written as special records
latest event is kept in logging process and fetched for display via corba

Alan Campbell CHEP'04

Calibration



L45 process
outputs ntuples of data needed to compute calibration
dumps to data stream as special records

l45const - inserts new constants barriers on run start if available on disk
or immediately on request via corba

Trigger Algorithm Steering

algorithm defined in text
stored in database

```
'MODULES;'
'L1=L_RNDM,L_L1_L5SKIP;'
'QT=I_QT_NHITCRJE,#L1;'
'CJC=I_CJC_NUPSTRTRACK,#QT;'

'L1RESET;'
'L_L1_ALLST      = 0-127;'

'TRIGGER;'
'* Accept Very High ETJET events;'
'L4_HS_VHETJET  =  L_L1_VHETJET                 &'
'                  R_CJC_ZVX>-60.               &'
'                  R_CJC_ZVX<110.               &'
'                  R_ENFL_ETJET>20.0            &'
'                  L_L1_SETSKIP                 &'
'                  L_L1_ECBIT07                 &'
'                           :ACCEPT:0.00:CONTINUE;'
'******* reject obvious background **********;'
'L4_CJC_VTXZ_DOWN= L_L1_ALLST                   &'
'                 R_CJC_ZVX>100.0:RESET_ALLST:0.1;'

'ACCEPT;'
'H1REC   :1.0;'

'HISTOGRAMS;'
'MAIN   = 50,0.,5000000.#    1,0;'
'QT     = 25,0.,  50000.# 1000,0;'
'L1     = 25,0., 100000.# 2000,0;'
'CJC    = 50,0.,2500000.# 4000,0;'
'R_FNC_ESUM = L4_EFS_FNC_Q2   :60, 10.,1210.;'
'R_CJC_ZVX  =                 200,-250.,150.;'
```

definition of processing modules
variables which can be calculated by module
dependencies between modules

definnition of trigger masks

statements executed until accept/reject
modules called only if variable needed
statement aborts if a subcondition is false
if all subconditions true action taken
ACCEPT/REJECT/RESET_MASK
RESET_MASK is reject only if no bits remain
CONTINUE specifies test statement
fraction specifies monitor/scaledown
histograms per statement records result

extra processing if accept

module timing histograms
variable histograms
variable histograms for individual statements

Alan Campbell CHEP'04

Summary & Conclusions

H1 Experiment High Level Filter

* Event Repository organises data flow and synchronisation
* multithreads overlap data transfer and processing
* successful mix legacy FORTRAN and with C++
* very stable system
* CORBA gives language & network independence
* entirely open standards, open source