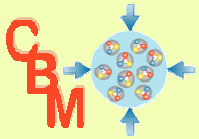


Developments with the XDAQ framework

J. Adamczewski, H.G. Essel, S. Linev

EE/GSI

Work supported by EU RP6 project JRA1 FutureDAQ RII3-CT-2004-506078

Outline

- Introduction
- Data transport (review, conclusions)
- Hardware driver integration with HAL
- Controls: Framework, XDAQ-DIM adapter
- Data flow modules in XDAQ
- Summary

Data Acquisition framework requirements

goal:„Data Acquisition Backbone Core“ DABC

- Modular architecture
- Data transport management
- Configuration of multiple nodes
- Controls, monitoring, message logging
- Error handling, failure recovery
- Hardware driver integration
- ...

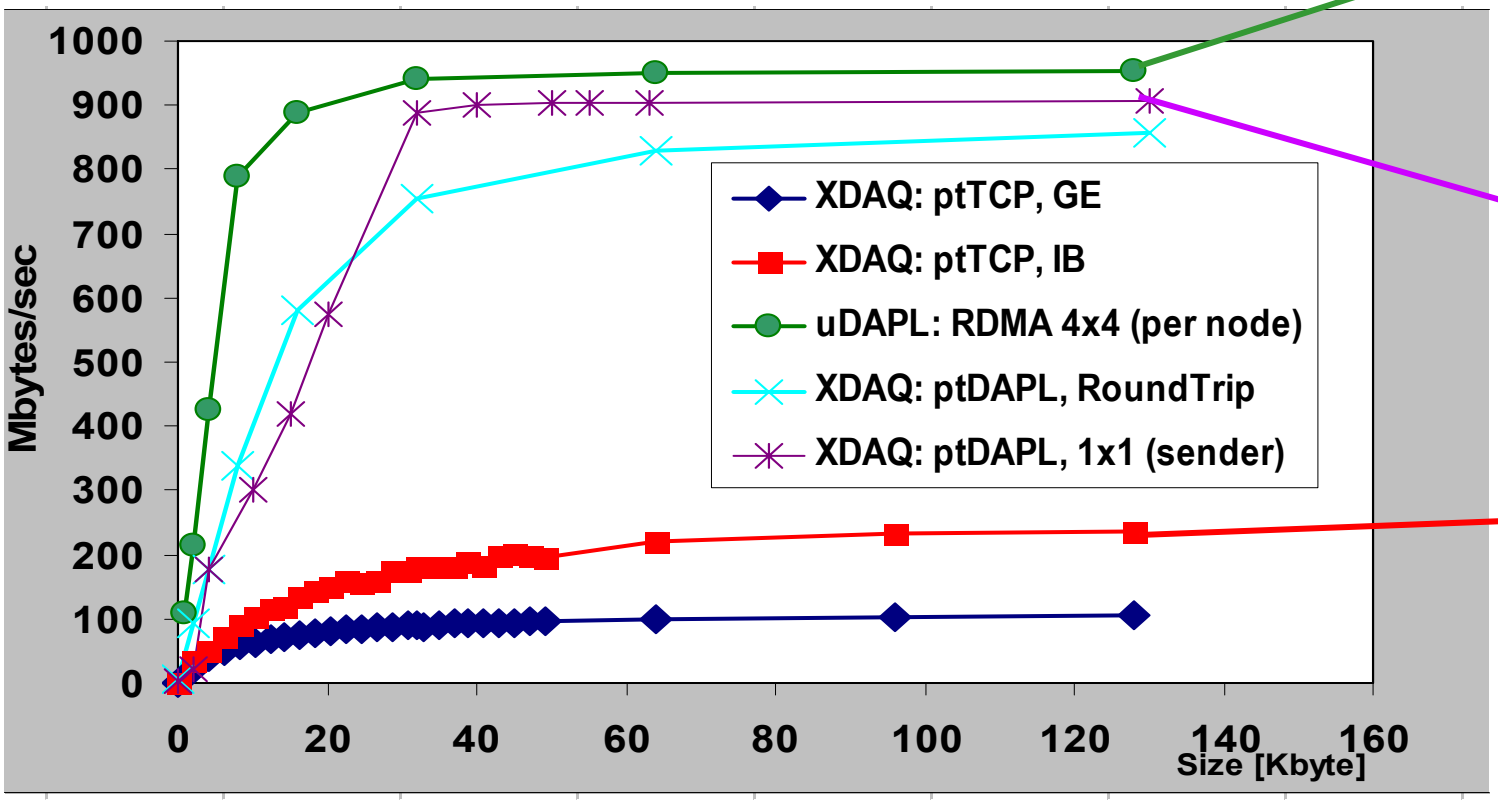
The CMS XDAQ framework

Standard DAQ framework for LHC CMS experiment

(Orsini, Gutleber) http://xdaqwiki.cern.ch

- C++ libraries on Linux, modular packages
- Process and thread management
- Cluster configuration: XML, application registry,...
- Data transport: Messenger+ peer Transport (I2O)
- Controls: state machines, infospace, http, SOAP,...
- Hardware Access Library
- ...

Data transport on IB test cluster (review)



955 MB/s

plain uDAPL

905 MB/s

XDAQ, I_2O, ptDAPL

234 MB/s

XDAQ, I_2O, TCP over IB

XDAQ Data Transport benchmark (review)

Peer Transport DAPL: GSI development 03/2006-05/2006

- **big packages**: network limit 950 Mb/s almost reached
- **small packages**: restricted by 15 µs framework latency

Problem: XDAQ peer transport is message oriented:

each package is dispatched separately by receiver id!

→ other Data Transport Interface for DABC...?

Hardware Integration to DAQ

Motivation:

Active Buffer Boards ABB (A.Kugel et al , Universität Mannheim) :

- deliver data from front end combiners
- PCIe
- FPGA for sorting and packaging
- Readout: DMA to PC memory

Other readout boards?

Control system hardware?

Hardware Integration with XDAQ (1)

XDAQ Hardware Access Library:

Software interface for communication from user space

- **BusAdapterInterface: bus protocol - VME, PCI, (PCIe)**
- **DeviceIdentifier: connection to kernel device driver**
- **AddressTable: ASCII or XML definition of (name, address) mapping**
- **HardwareDevice: uses the above + board specific functions**

Problem: generic PCI implementations of XDAQ release 3.6 not usable (linux kernel 2.4 only, no 64bit, no PCIe!)

=> Developed new implementations based on generic PCIe driver of Technische Informatik Uni Mannheim

(thanks to G.Markus, H.Singpiel, A.Kugel)

Hardware Integration with XDAQ (2)

New HAL classes:

- PCIFDaqBusAdapter:
- PCIFDaqDeviceIdentifier: wraps Mannheim driver C++ interface

Test example: standard GSI PCIGTB2 board (J.Hoffmann, W.Ott)

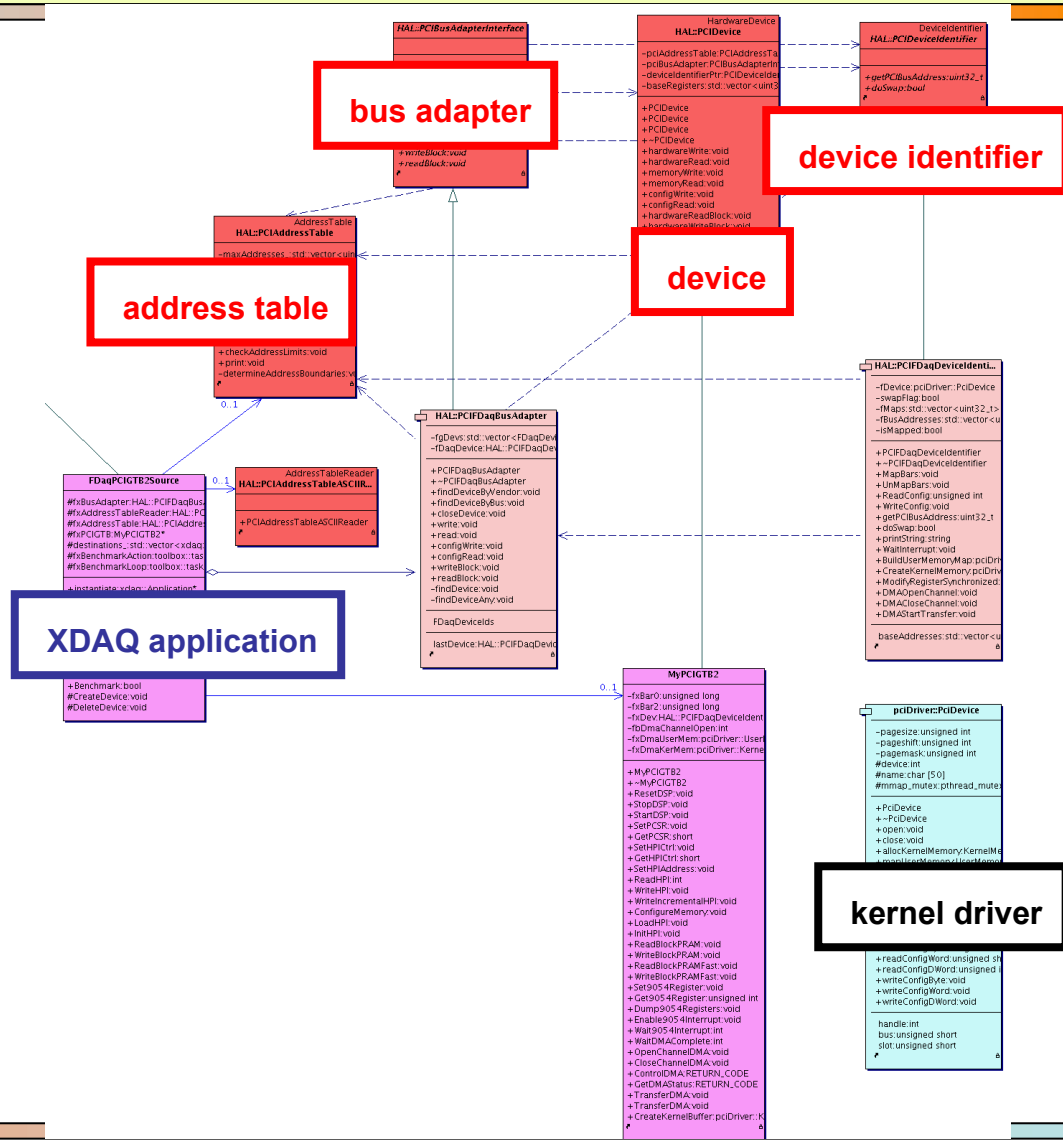
plain PCI; DSP registers; PRAM i/o ;(DMA?)

- MyPCIGTB2: implements HAL::HardwareDevice
- defined ASCII device address table
- kernel module modifications:
  - added device id, vendor id (for convenience)
  - board specific functionality (DMA, IR handler) for testing only!

Hardware Integration with XDAQ (3)

Class diagram:

- **HAL interface**
- **Mannheim PCI driver lib**
- **Implementations for Mannheim driver**
- **PCIGTB2 device**



Hardware Integration with XDAQ (4)

Tests with PCIGTB2 board (thanks to J.Hoffmann and W.Ott) :

- Hardware access via HAL from test console or XDAQ application (dummy readout)
- Successful:

  PCI addressing, load and execute DSP programs, bus r/wr on PRAM,
- Still failed:

  DMA from board PRAM (maybe hardware problem...)

Experiences:

- Integration of Mannheim driver is working! XDAQ is ready for AB Board
- generic PCI addressing: timing problems for special situations (PCI hang up!)
- generic PCIDevice::memoryRead(): Performance loss at PRAM block IO (<= 30%)
- No generic HAL interface for DMA and interrupt handling!
- Possible to implement advanced/optimized functionality in PCIDevice subclass!

DAQ Control system

XDAQ control system features:

- **State machine classes**

  free definable, transition functions, event driven, (a-)synchronous

- ***Infospace*: set of process variables**

  (de-)serializable from/to XML or SOAP, access by name

- **Webserver on each process *Context* (nodename:port)**

  any browser as user interface; http requests

- **SOAP messaging**

  Java (-script) controls GUI, *XRelay* fan out, infospace monitoring

Controls framework for XDAQ

Designed class hierarchy for all XDAQ applications:

FDaqApplication

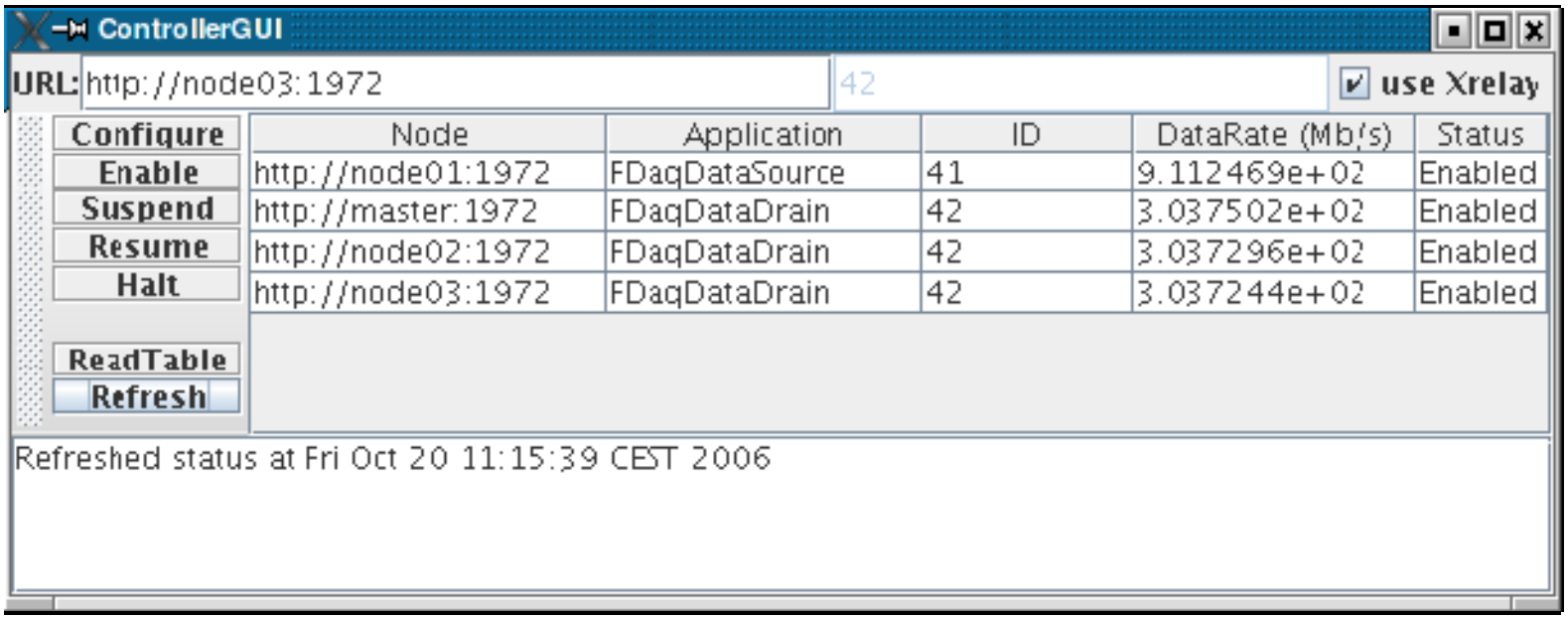
<- FDaqDataNode

<- FDaqDataSource, FDaqDataDrain, FDaqRoundTrip,...

FDaqApplication provides common controls features:

- state machine with command definitions
- process variable set (*XDAQ infospace*)
- interfaces: web server, SOAP, plain http request
- DIM server (see below!)

JAVA Control GUI study (ECLIPSE, SWING)



- **SOAP (de-)serialising well supported by XDAQ and JAVA**
- **Native Swing widgets sufficient?**
- **GUI needs active request to FDaqApplication for status refresh**

  (command-response mechanism instead publisher-subscriber)

Distributed Information Management DIM

- developed at CERN (C.Gaspar)
- C, C++, and JAVA libraries
- publisher-subscriber model:

  sends changed values to all subscribed clients
- handles free definable client commands
- CS-Labview-DIM interface (D.Beck)

  for Labview controls GUI
- EPICS-DIM gateway under development (P.Zumbruch)

DIM server in XDAQ application

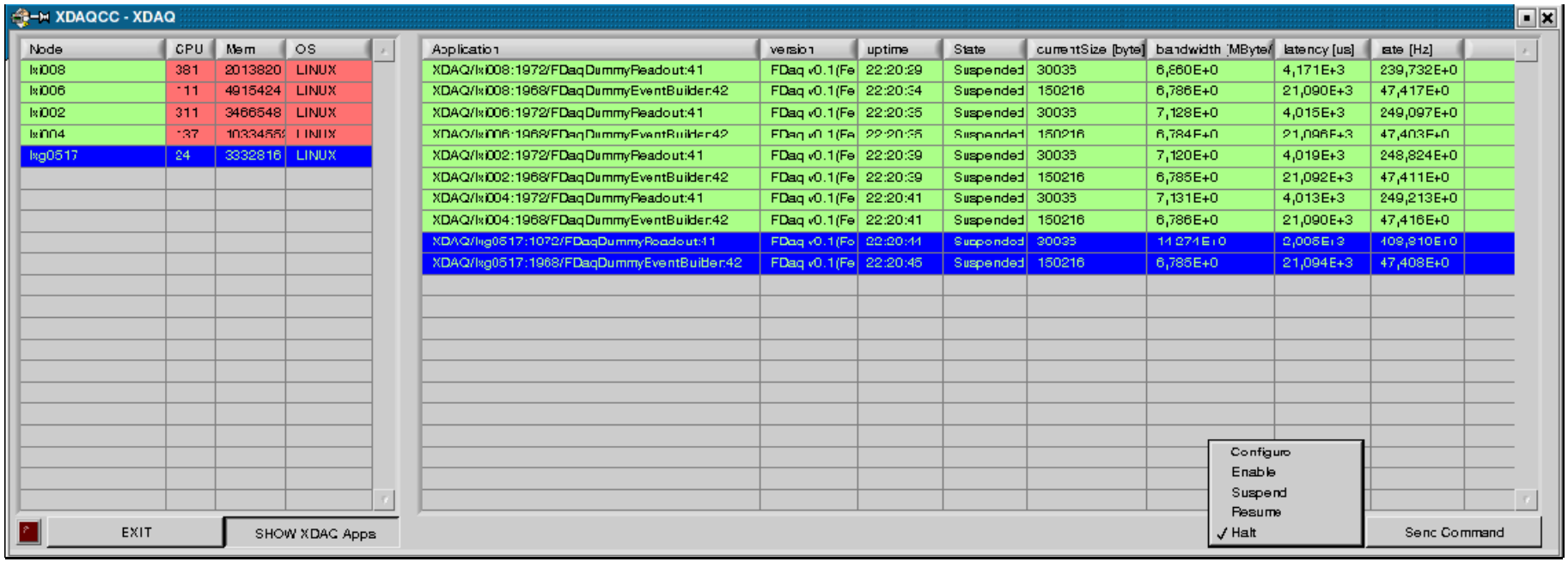
FDaqDimServer adapter class:

- subclass of DimServer (DIM C++ interface)
- singleton for all FDaqApplications in XDAQ process
- dispatches DIM commands to FDaqApplication callbacks

FDaqApplication base class:

- exports all XDAQ infospace variables as DIM services
- XDAQ “variable change event” will trigger DIM service update
- DIM commands for state machine switching
- DIM commands for parameter changing (if allowed)
- special DIM service for “heartbeat” update (state, uptime)

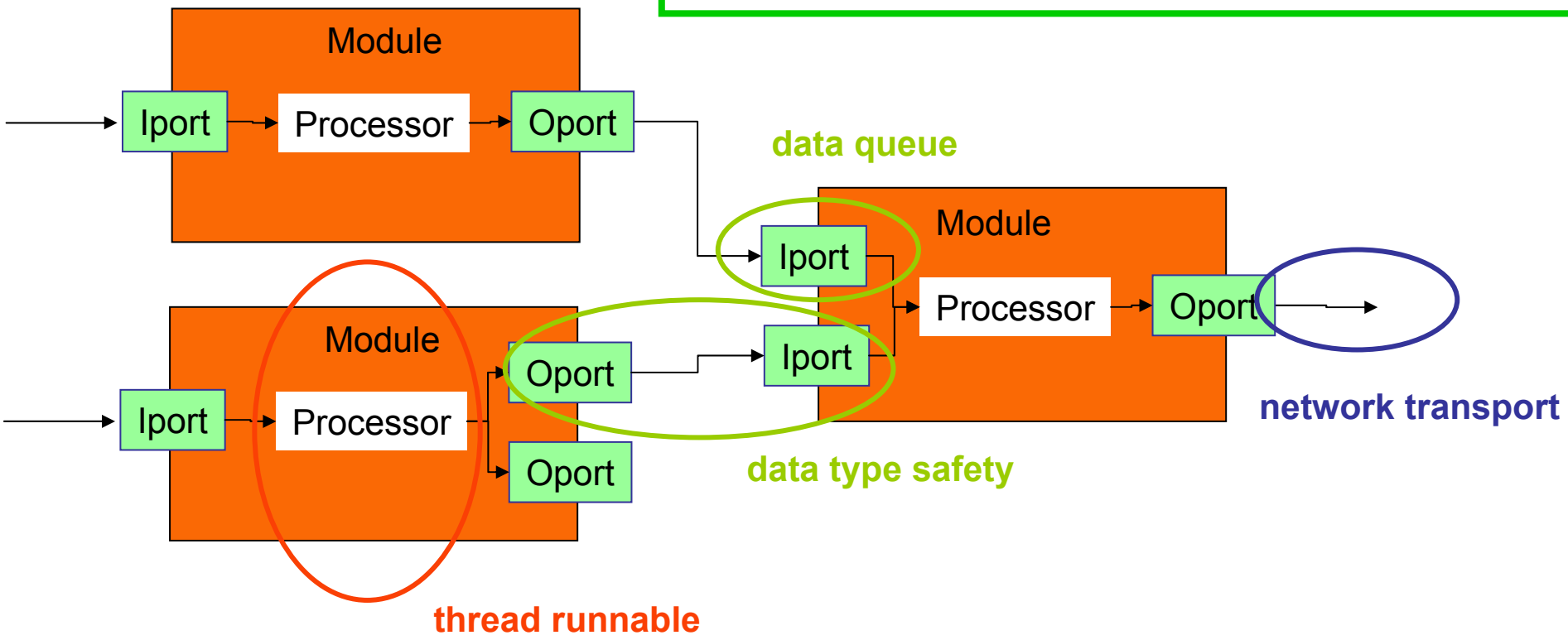
Labview GUI for XDAQ DIM server



- **Uses Labview-DIM interface and CS nodemon**
- **XDAQ application list is set up dynamically from DIM service**
- **Displays common parameters of all applications (state, data rates)**
- **Many thanks to Dietrich Beck who made this Labview VI within 2 days...**

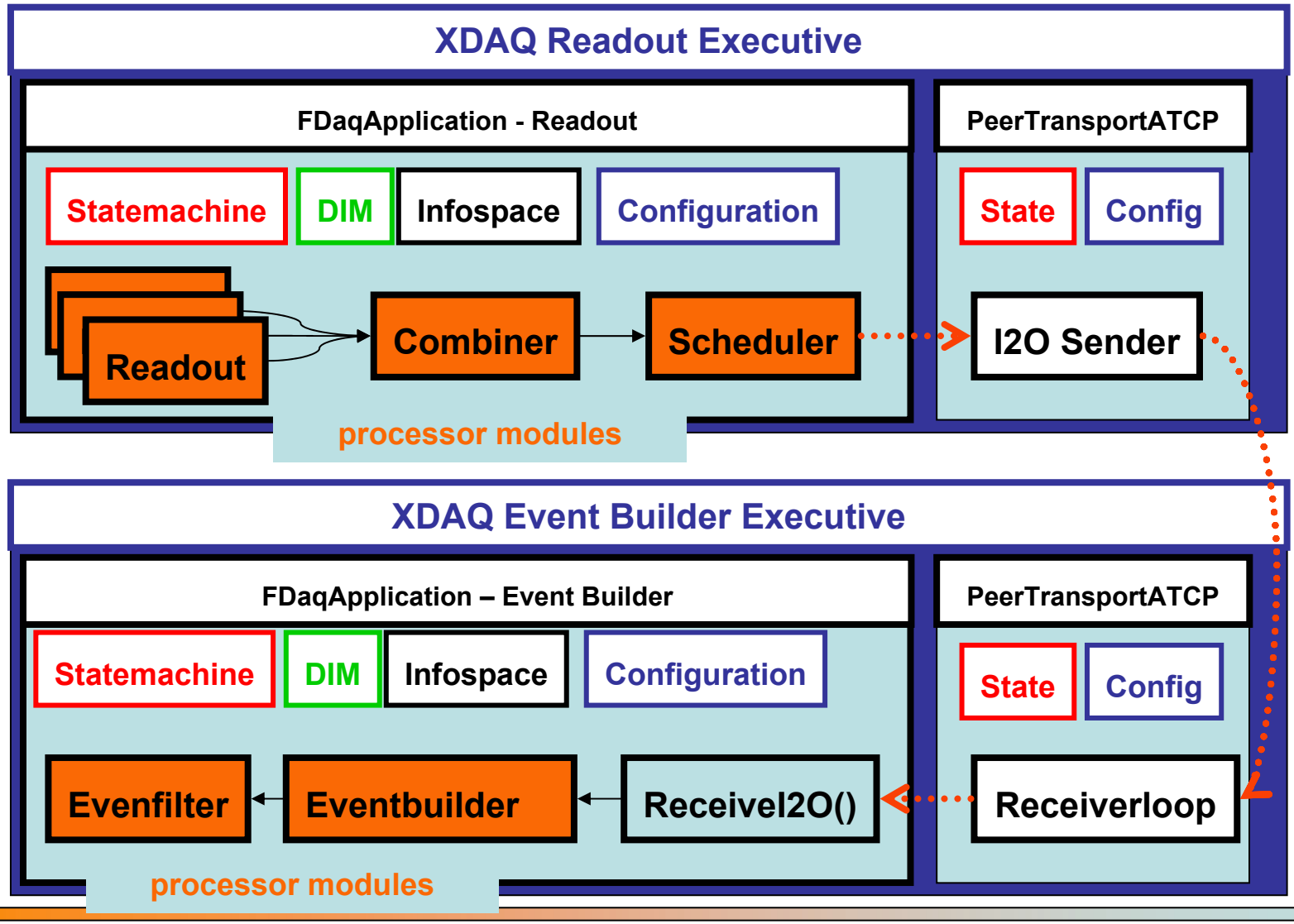
DABC: dataflow between modules

See talks of H.G.Essel and S.Linev



configuration? connection management? run control? error handling?...

Dataflow modules example with XDAQ



n x m

B-network

Summary and Outlook

XDAQ PeerTransport interface:

- performance loss (latency) by message oriented approach
- To do: develop connection oriented transport layer! (DABC)

XDAQ Hardware Access Library:

- Integration of ABB PCIe device driver done.
- To be implemented: DMA, interrupt service, FPGA access

  (need the ABB board with finished kernel driver to do this!)

Controls framework:

- Developed controls class hierarchy for XDAQ applications.
- Developed DIM server adapter for XDAQ applications.
- First Labview GUI for XDAQ via DIM tested; to be improved.

DABC dataflow modules:

- First XDAQ implementation tested on 5 nodes; under development!