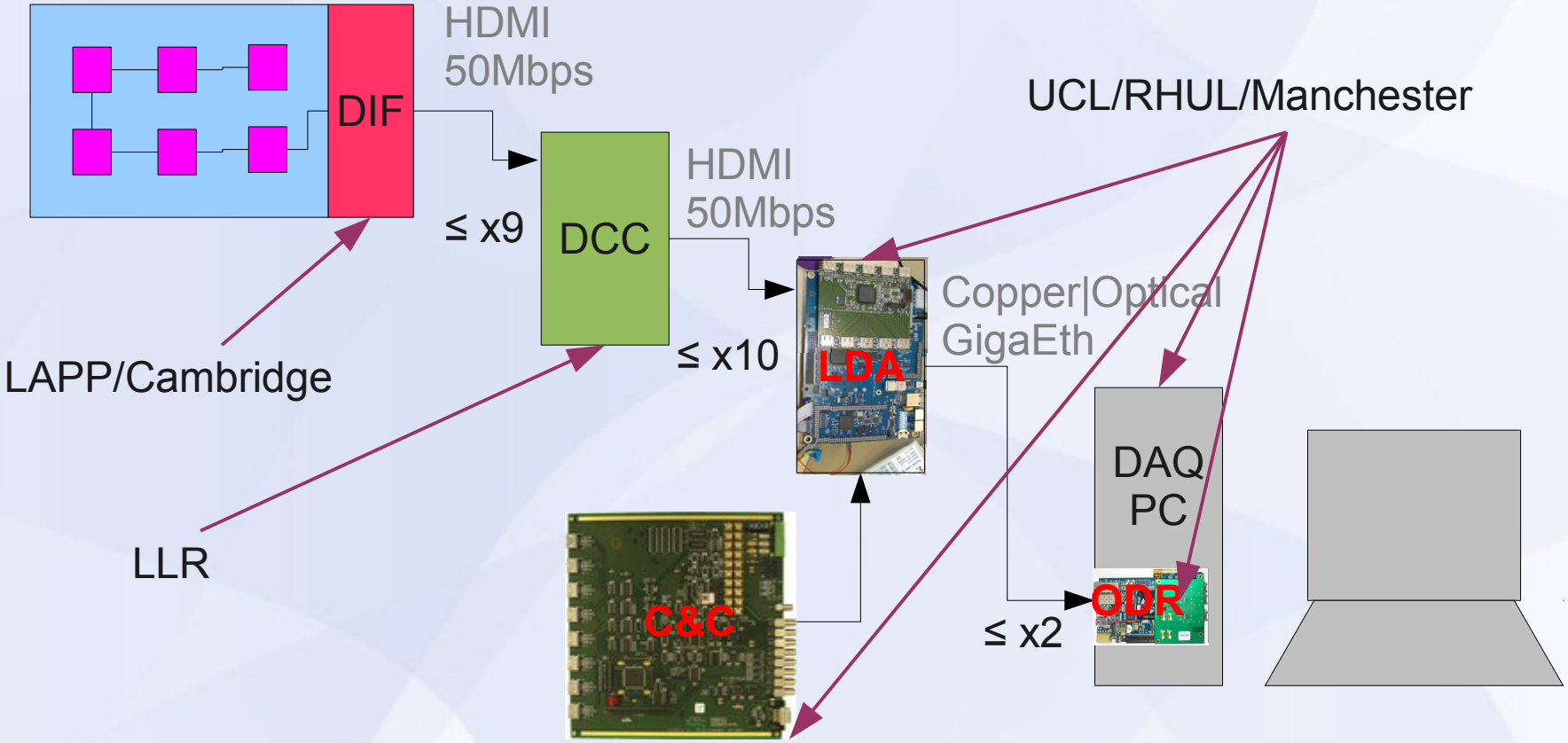
DAQ2 Calice: Software

David Decotigny
With Rémi Cornat and Franck Gastaldi for 2010 LLR Agenda

Outline

- Software status
- Hardware and Software agenda @LLR
- A few open questions

“DAQ2” Calice: System Overview



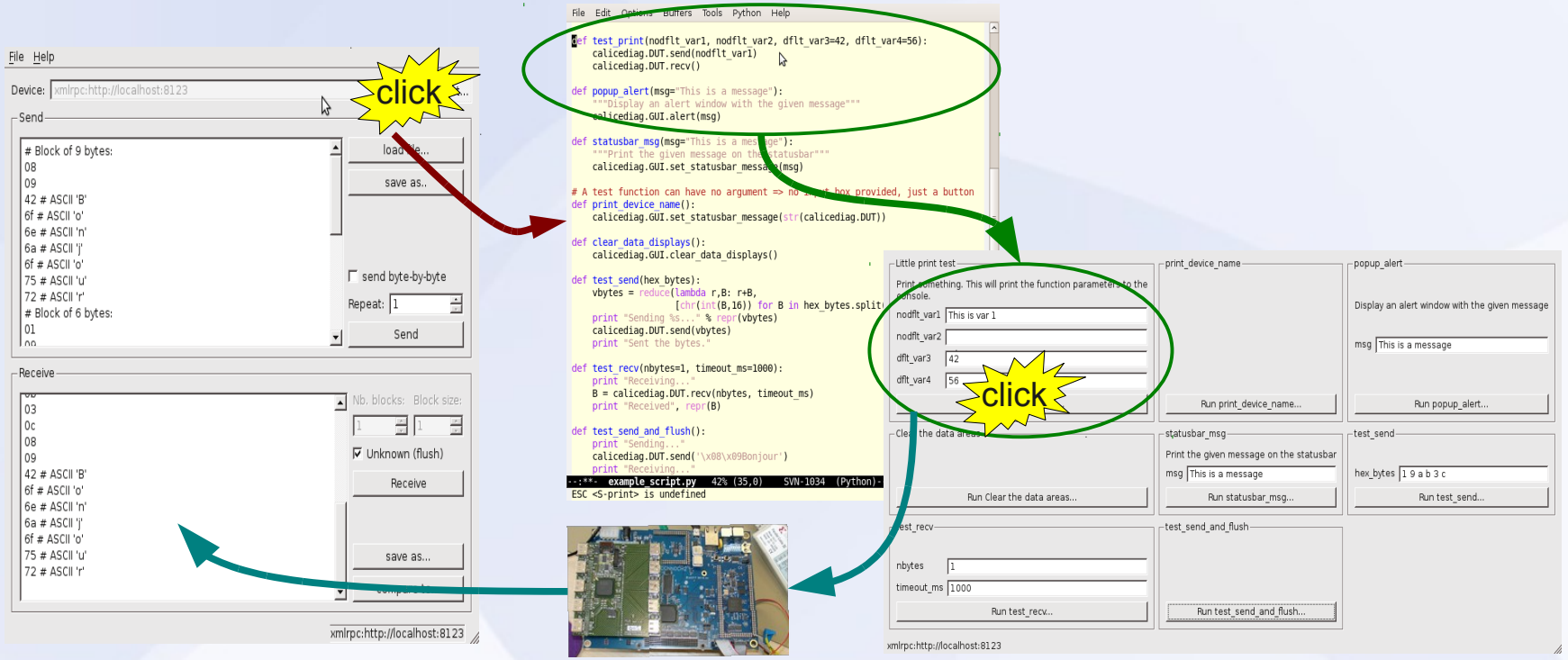
A network-oriented setup: a tree of data concentrator cards

Software status
Hardware/Software interfaces

- ODR kernel driver by A. Misiejuk (Manchester)
  - Able to store data packets to disk
- CCC interface (DOOCS + Agnostic)
  - Register accesses
- LDA interface (Agnostic)
  - Register accesses through ethernet
- DCC interface
  - Tests through USB
- DIF interface
  - Tests through USB
- Integrated DAQ Software (DOOCS, V. Bartsch)
  - GUIs to control the ODR + CCC
  - FSM to control the device servers
  - DB framework to retrieve configs

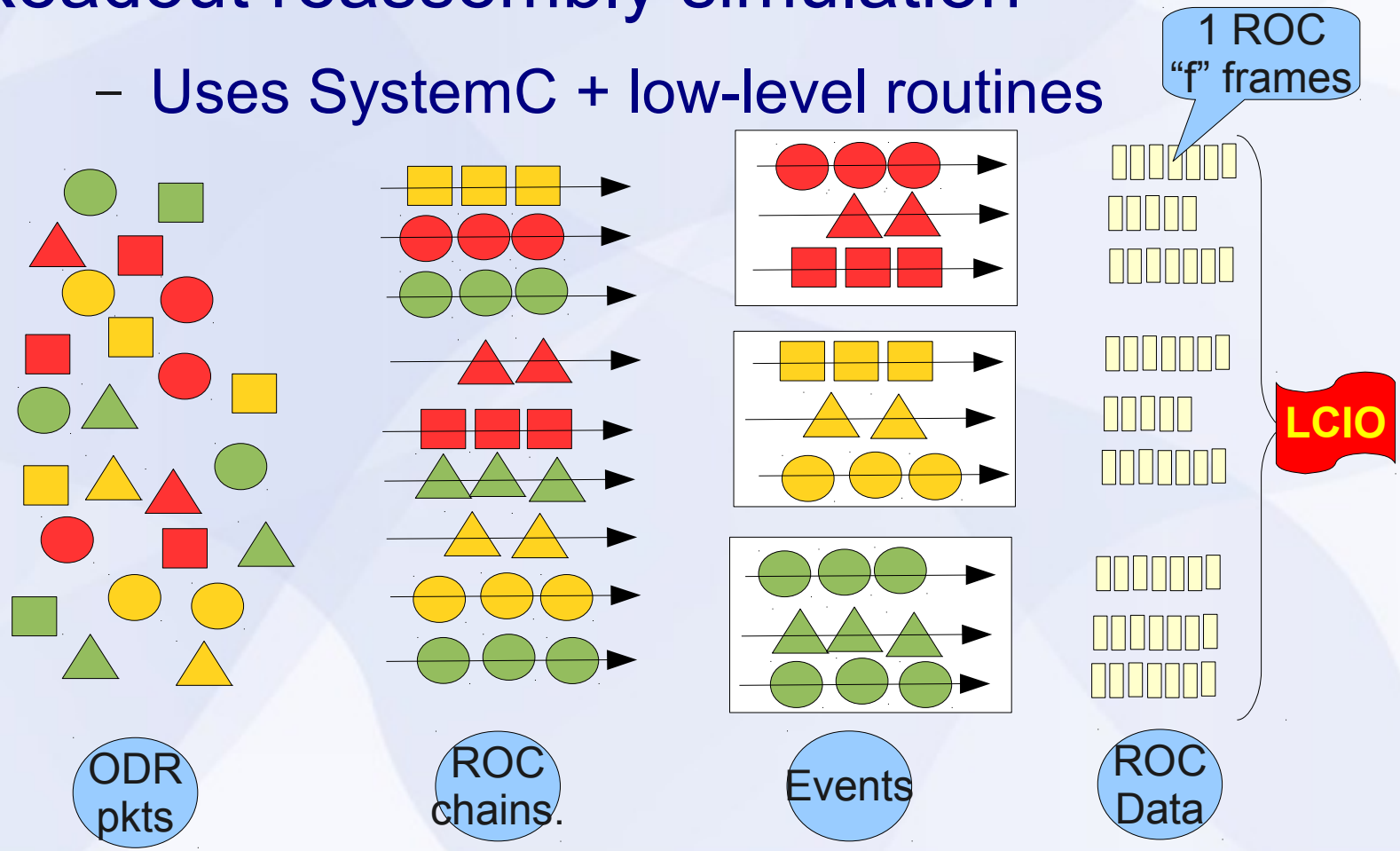
Software status
Hardware debugging tools

- `pyserdiag`: customizable tool to talk with USB/RS232/Ethernet under Windows+Linux
  - https://svn.in2p3.fr/calice/online-sw/trunk/pyserdiag/



Software status
Other software

- Readout reassembly simulation
  - Uses SystemC + low-level routines



2010 Hardware/Software agenda @LLR

- March'10:
  - HW: DIF working with a DCC board
    - Serial link tested and validated
    - Part of specifications of the HDMI side implemented
    - USB I/O of the DCC
  - SW: Reassembly algo
    - Evaluation & optimization of reassembly algo with real PC hardware & network links from simulated Data
    - Support for HW developments

2010 Hardware/Software agenda @LLR

- June '10:
  - HW: DIF working with a DCC + LDA
    - (almost) full specifications implemented
    - Including ROC chips functions (from Guillaume/Mathias)
    - Ethernet I/O of the LDA
  - SW: LDA+DCC+DIF support
    - Start of SW layer for DAQ2 with LDA and Guillaume's firmware, interaction with C. Combaret & L. Mirabito (Xdaq)
    - Device Servers for Tango and/or DOOCS
    - Ongoing support for HW developments

2010 Hardware/Software agenda @LLR

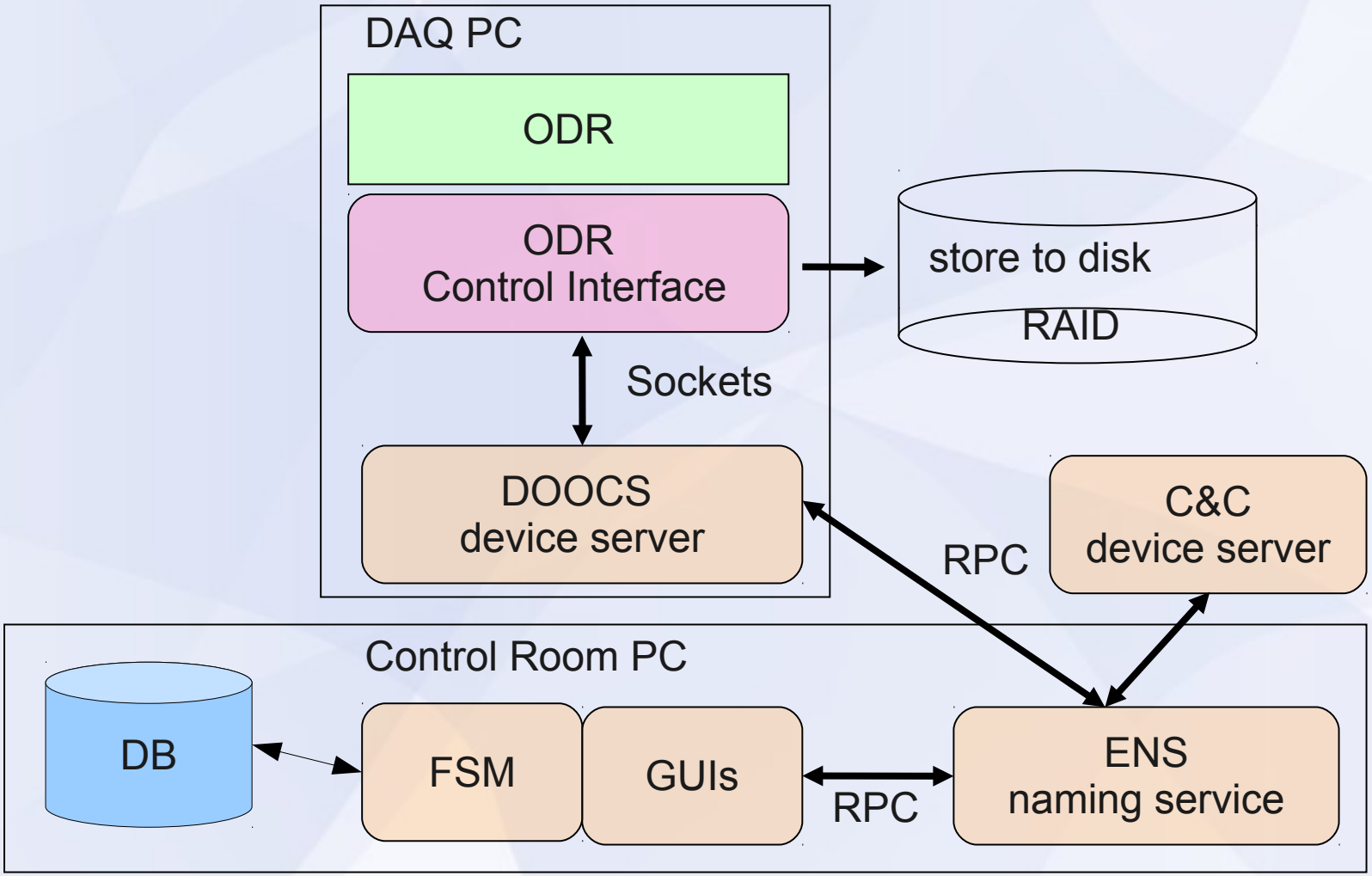
- September '10: DAQ2 SW+HW converge
  - HW: more than 1 DIF working with a DCC + LDA
  - SW: DAQ2 support for DHCAL testbeam
    - With C.Combaret/L.Mirabito: Integration of the DAQ2 Software for m^2 DHCAL
    - Generic DAQ: Integration with ILC Soft ?
- Later on:
  - Validate integration of HW/SW in testbeams
  - Integration with machine infos/beam infos
  - Integration with other subdetectors

Open questions

- Common data pipeline across subdetectors ?
  - Common scheme (load balancing to several servers, how ?) and data format ?
- Common configuration database ?
  - Common way to configure devices (same FSM ?) ?
- Common way to identify coming from the same event across subdetectors ?

Backup slides

DAQ2 SW architecture



+ Simulation algo reassemblage (Read-out)

DAQ2 : Perspectives et besoins

- Début 2010 :
  - Integration/debug de la DAQ anglaise + DCC
    - Programmation des DIFs, DCC
  - Mesurer les performances de l'algo de réassemblage
- Mi-2010 et suite :
  - Soft DAQ réutilisable
    - Xdaq ou Tango pour slow-control
    - EUDAQ pour read-out/stockage/integration avec autres DAQ
  - Intégration dans une DAQ commune (HW+SW) :
    - Intégration avec ILCSoft
    - Gestion trigger unique : TLU + C&C
    - Récupération caractéristiques faisceau : “BIF” ?
    - Récupération caractéristiques machine

DAQ2 : Perspectives et besoins

- mi-2010
  - Installation d'un banc DAQ2 à l'IPNL et/ou au LAPP pour test/debugging (mat donné par UK)
- Q3 2010
  - test en faisceau avec plusieurs plans

Prévision des besoins

- LLR :
  - Matériels : PC, carte ethernet optique
  - Efforts de développement :
    - Courant 2010 :
      - debug/mise au point du coeur de la chaîne avec les électroniciens
      - slow-control/read-out simple (Xdaq/Tango, EUDAQ/maison)
    - À partir de fin 2010 : **renfort manpower souhaité**
      - intégration avec ILCSoft
      - Intégration avec les autres détecteurs (testbeams combinés ; ex. Support CCC+TLU)
      - Récupération des informations de la machine
      - Récupération des caractéristiques faisceau (“Beam InterFace card” ?)

Prévision des besoins

- IPNL
  - ??
- LAPP
  - Production des DIF (ANR)

Bonus slides

Slow-control : Choix retenus

- Embarras du choix pour les SCADA pour HEP:
  - DAQ1, Xdaq, Tango, EPICS, Labview, PVSS, MIDAS, home-made, …
- Plusieurs utilisés dans ILC
- Pour Calice:
  - Xdaq+Labview par C. Combaret à l'IPNL
  - DAQ2: Évaluation Xdaq + Tango (ESRF)

Data-readout : choix retenus

- Un peu moins de frameworks disponibles:
  - EUDaq, DAQ1, Xdaq, ICE, Narval, home-made (file/socket-based), …
- Plusieurs utilisés dans ILC:
  - Exemples: Xdaq, EUDaq, DAQ1, Narval, home-made
- Pour Calice:
  - Xdaq par C. Combaret à l'IPNL
  - DAQ1: “simple serialization” par P. Dauncey
  - DAQ2: EUDAQ pour interfaçage avec les autres ? Ou maison
    - Tenter d'aller jusqu'à sauvegarder en LCIO (éval.)

Prévision des besoins

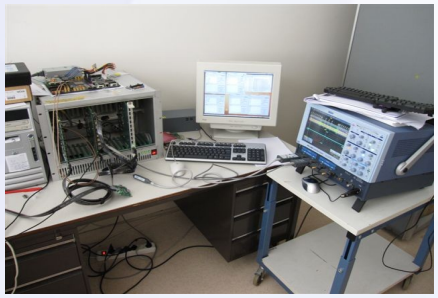
- Matériels : PC, carte ethernet optique
- Efforts de développement :
  - Courant 2010 :
    - debug/mise au point du coeur de la chaîne avec les électroniciens
    - slow-control/read-out simple (Xdaq/Tango, EUDAQ/maison)
  - À partir de fin 2010 : **renfort manpower souhaité**
    - intégration avec ILCSoft
    - Intégration avec les autres détecteurs (testbeams combinés ; ex. Support CCC+TLU)
    - Récupération des informations de la machine
    - Récupération des caractéristiques faisceau (“Beam InterFace card” ?)

Status Software

- Prototype tournant sous DOOCS
  - ODR kernel driver par A. Misiejuk (Manchester)
    - Capable d'enregistrer les paquets de l'ODR sur disque
  - Interface CCC pour DOOCS
    - Accès registres
  - Software DAQ par V. Bartsch (UCL)
    - GUIs de contrôle de ODR + CCC
    - FSM de contrôle des device servers
    - Squelette DB pour les configs

Hardware/Firmware status

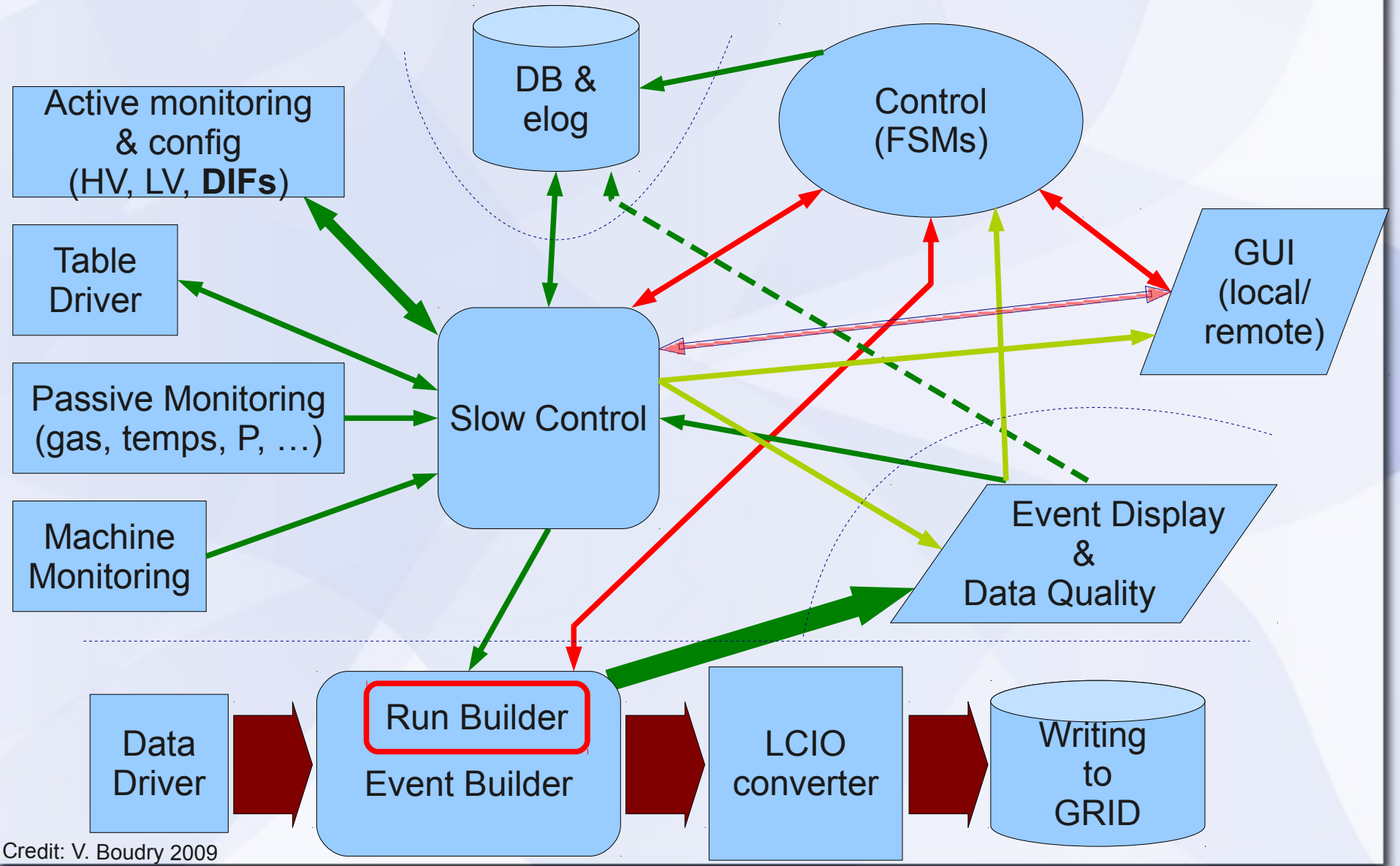
- LDA+ODR+C&C+DAQ PC:
  - Hardware Provided to LLR by UCL/RHUL/Manchester
  - Firmware developed by UCL/RHUL/Manchester



- DCC:
  - Hardware available @LLR
  - Firmware developed by F. Gastaldi

- DIF:
  - Several prototypes available @LLR
  - DIF Task Force for the firmware

Perspectives : mi 2010 et après



Credit: V. Boudry 2009

Slow-control/supervision Preparing for integration issues

- (Some other) slow-control subsystems to consider, same integration problems:
  - Configuration
  - System dynamics (FSM)
  - Error handling, logging
  - Online monitorimg: rendering with ROOT ? FROG ? Other ?
  - Trending database
  - ...

Data-readout path
Preparing for integration issues

- (Some) other aspects to consider, new integration issues:
  - Correctly identify readout data coming from different subdetectors as belonging to the same physical event
    - Uniquely identify each trigger ? Specialized hardware (TLU/CCC ?)
    - P. Dauncey: order the readout phases...
      - Need to evaluate this solution for network-oriented DAQ

Data-readout path
Preparing for integration issues

- (Some) other aspects to consider, new integration issues (ctd.):
  - Data volumes: single machine able to store ?
    - Specific hardware needed ?
    - “Intelligent” load-balancing switches to transfer data to PC clusters ?
    - Zero-suppression, selective readout, on-the-wire lossless compression ? Other ?
    - ➔ Not really a concern for Calice... how about the other subdetectors ?
  - Storage data format ?

Monitoring the beam

- To determine the accurate actual characteristics of the beam
- Usually: readout of Scintillators, Cerenkov, MWPC, fiber hodoscopes, etc. specific and/or supported by the hosting testbeam facility
  - Need to adapt the setup for each testbeam location
- "Beam Interface Card" (BIF): common hardware to monitor the beam
  - Proposals by V. Boudry, R. Cornat, F. Gastaldi (LLR) and J. Prast (LAPP)
  - Hardware card + associated driver software
    - Compatible with the ILC acquisition modes
      - Including auto-triggered: would allow to have an accurate insight into the beam characteristics for each recorded event
    - Readout of a wide range of signals used by the devices above

A BIF proposal

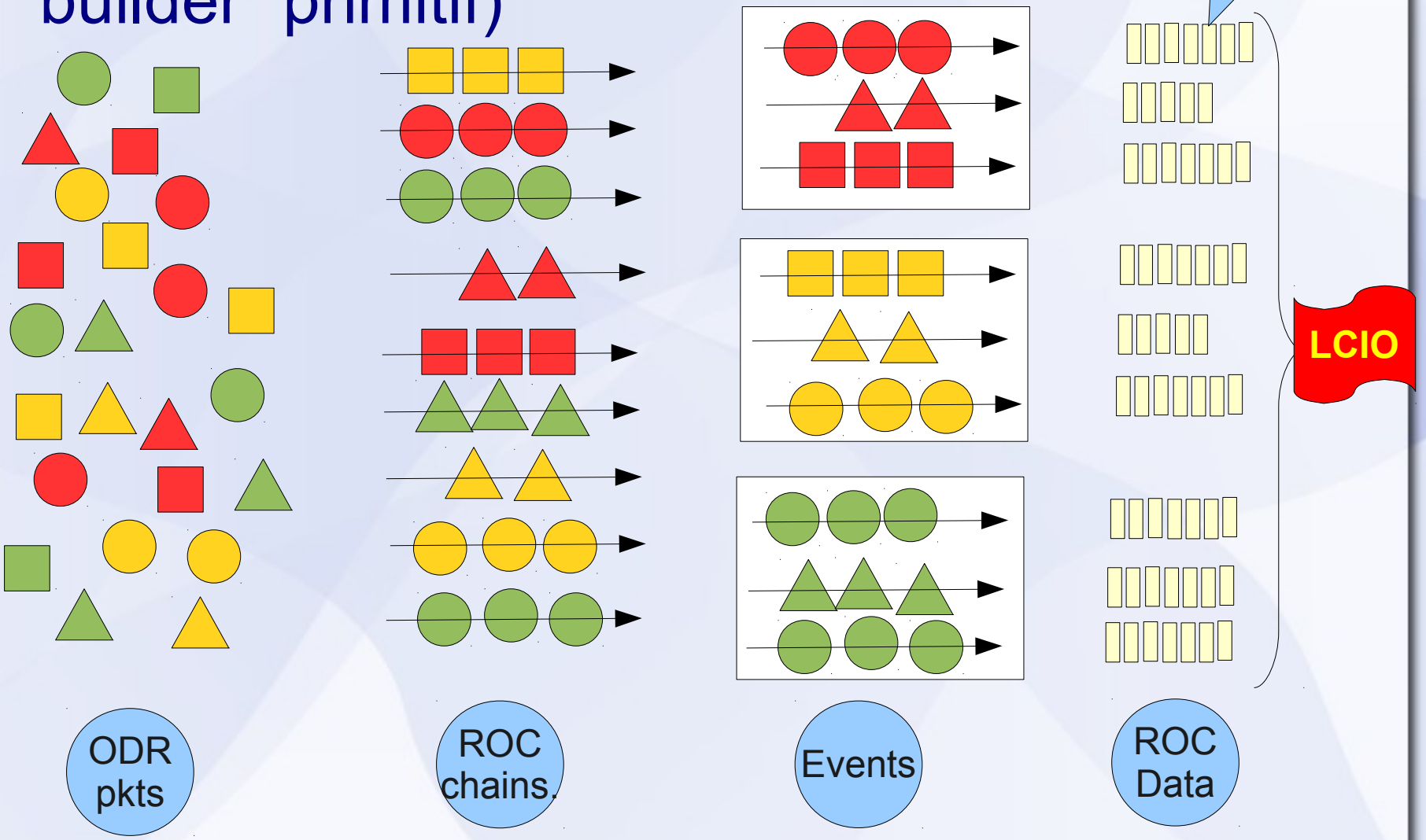
- A ROC-based solution:
  - 1 or 2 ROC receive the external signals and store them
  - Pros/cons:
    - ROC Number of channels seem large enough (eg. 10 scint, 16 C PM, 8 MWPC)
    - SPIROC stores time+amplitude: needed for MWPC
    - Memory size might be too short for DHCAL
      - use 1 HR for digital signals (Scint, C PM), 1 SPIROC for MWPC ?
    - Trigger logic provided by the outside world
  - Readout by a standard DIF
  - More demanding in hardware design manpower

Another BIF proposal

- A DIF-based solution
  - Adapter board + modified DIF
    - Existing hardware, FPGA-based
  - Pros/cons:
    - Cheap+simple adaptation board (~ connectors)
    - Large memory needed onboard FPGAs
    - No amplitude recording
    - “Easily” reconfigurable trigger logic
  - More demanding in VHDL manpower

Réassemblage de paquets

- Simulation réassemblage (“event builder” primitif)



Interfacing with the machine
Monitoring the environment

- To get get knowledge about:
  - beam energy & particle type (⟷ magnets current & upstream collimators position)
  - beam intensity ("counting devices")
  - final collimators position
  - beam profile (monitoring chambers histograms)
  - Cherenkov pressure
  - and others: existing environmental measures (P, t, humidity), …
- Have a common API for all the machines ?
- All this with fast access (≤ 30s)
- An open question ?

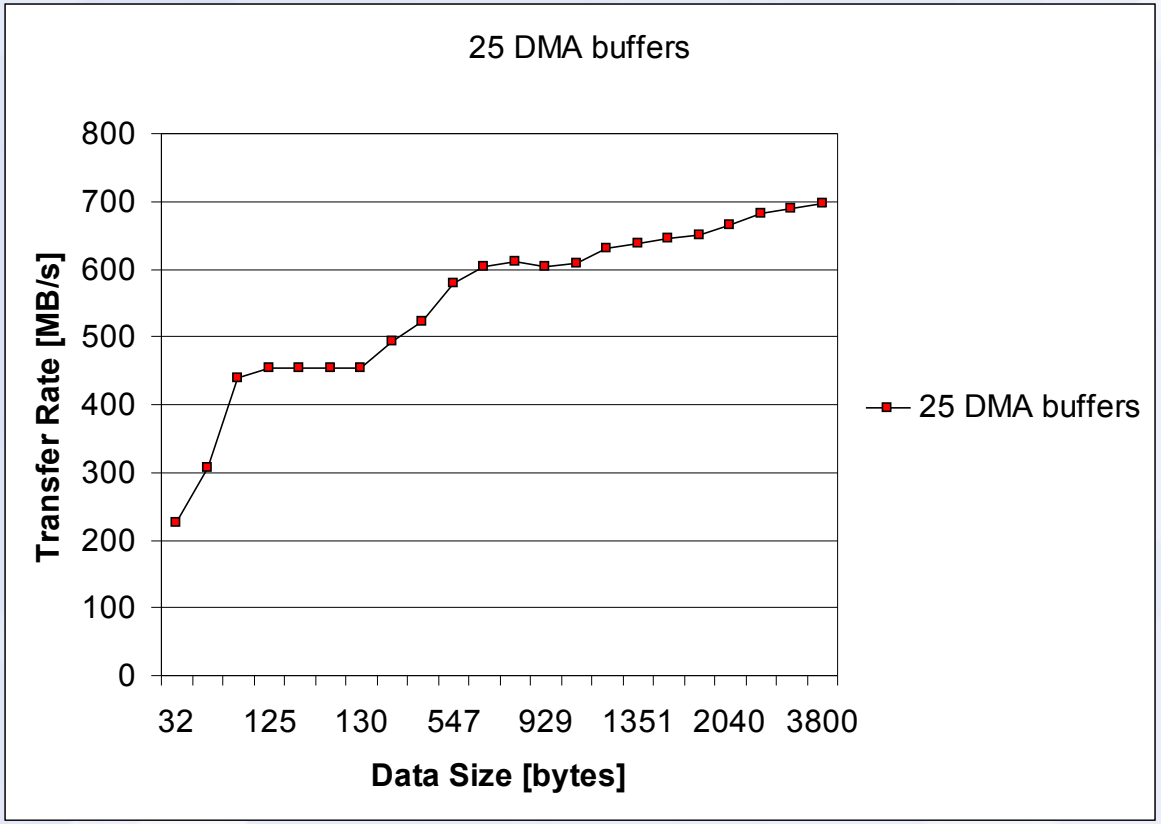
DAQ Framework: Tango ?

- http://www.tango-controls.org/
- Used by many synchrotron exp. (ESRF, DESY, Soleil…)
- But: the base is NOT synchrotron-specific
- Generic DAQ distributed system (slow-control)
  - Distributed configuration stored in DB
  - Service interaction through an ORB (~ RPC bus)
- Nothing really technically original, but more modern, nicer, more mature, less DESY-centric than Doocs

DAQ Framework: Tango ?

- The things that made me enthusiastic:
  - A real 3rd-party blob: we are *users* of the framework, we don't go *inside* the framework
  - Simple generic & composable message data types between services
  - Reasonably language-agnostic: C++, java, python
  - Can use jddd (Doocs GUI builder)
  - Management can be achieved via GUIs
  - A fast “device server design” approach (GUI code-generator)

DAQ Prototype: Performances



Evaluation by V. Bartsch/T. Wu/UCL/Manchester