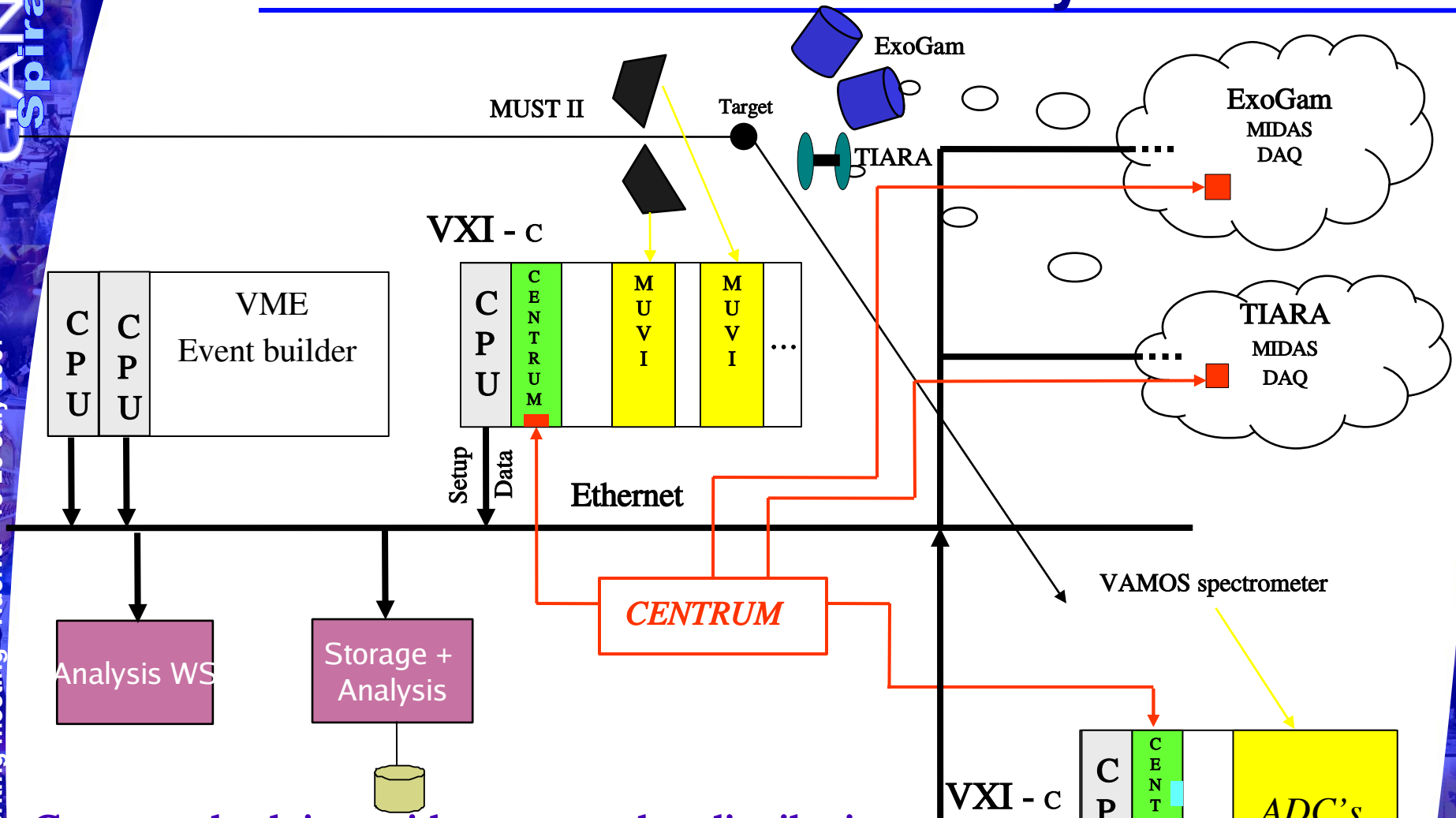
Bruno RAINE

GANIL-Caen

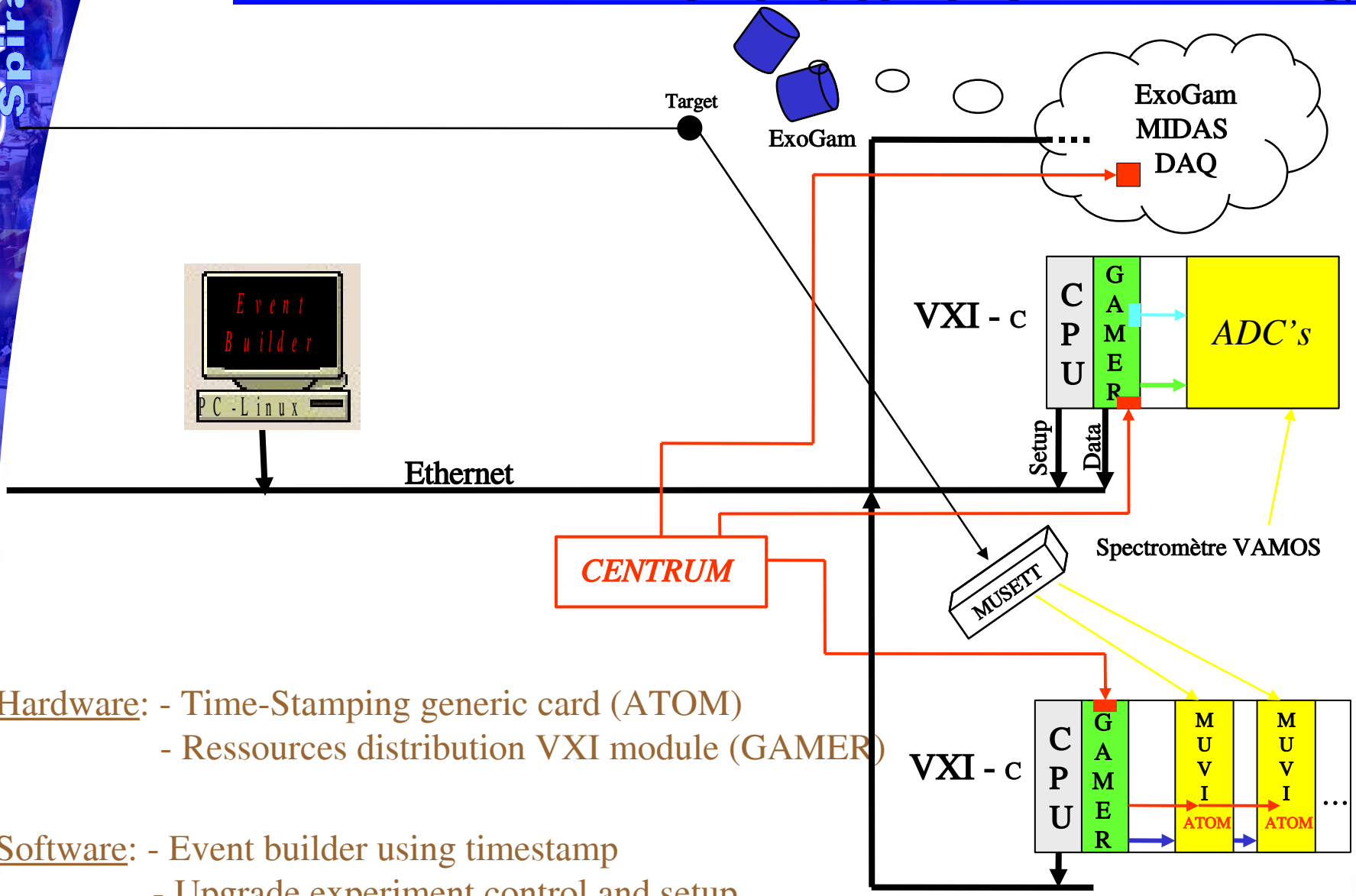
Future of the Ganil DAQ ?

Today GANIL DAQ



- Common dead time with event number distribution
- Independent setup for each subsystem
- Run control overall the system
  - selection of active detector, start, stop, status, monitor data rate

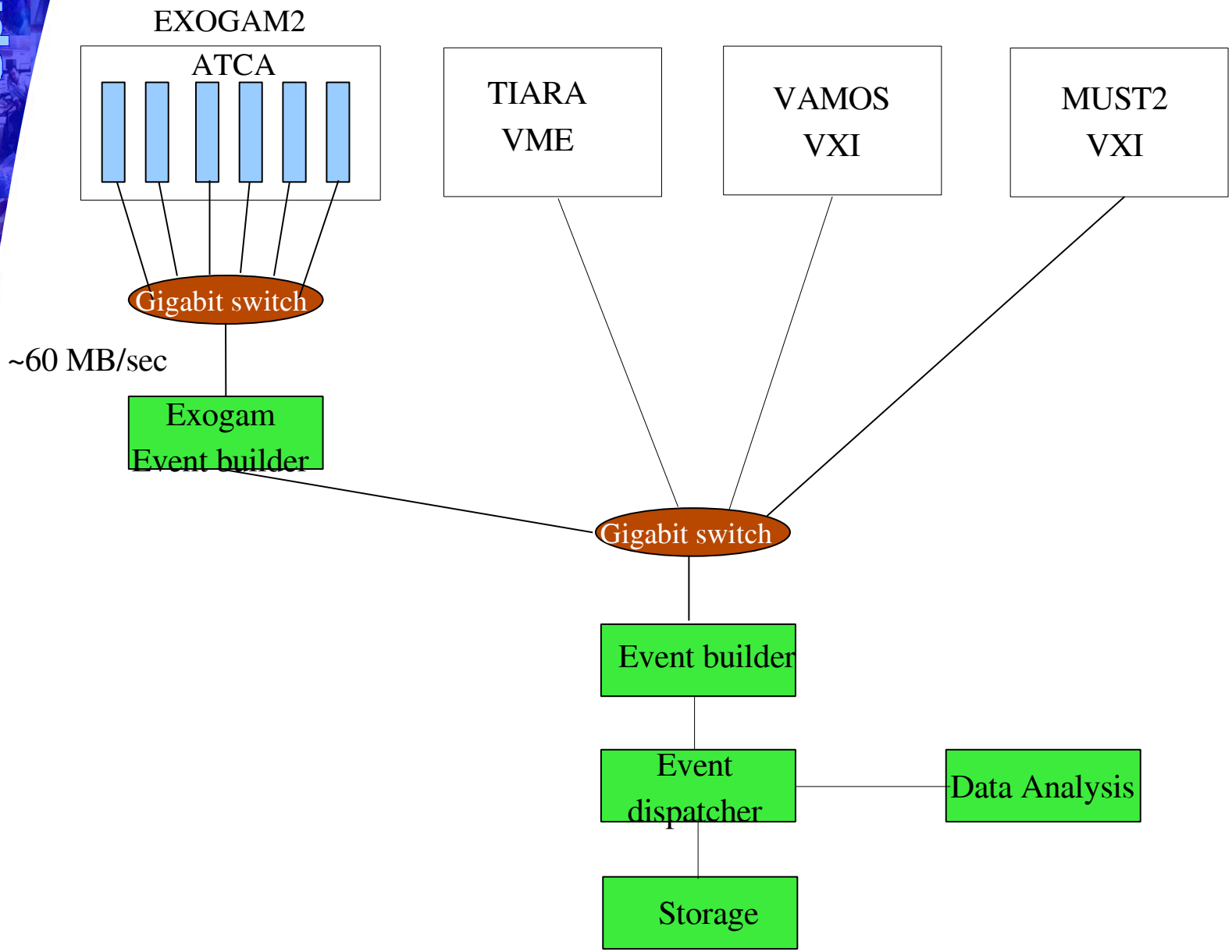
Next future GANIL DAQ



Hardware: - Time-Stamping generic card (ATOM)
- Ressources distribution VXI module (GAMER)

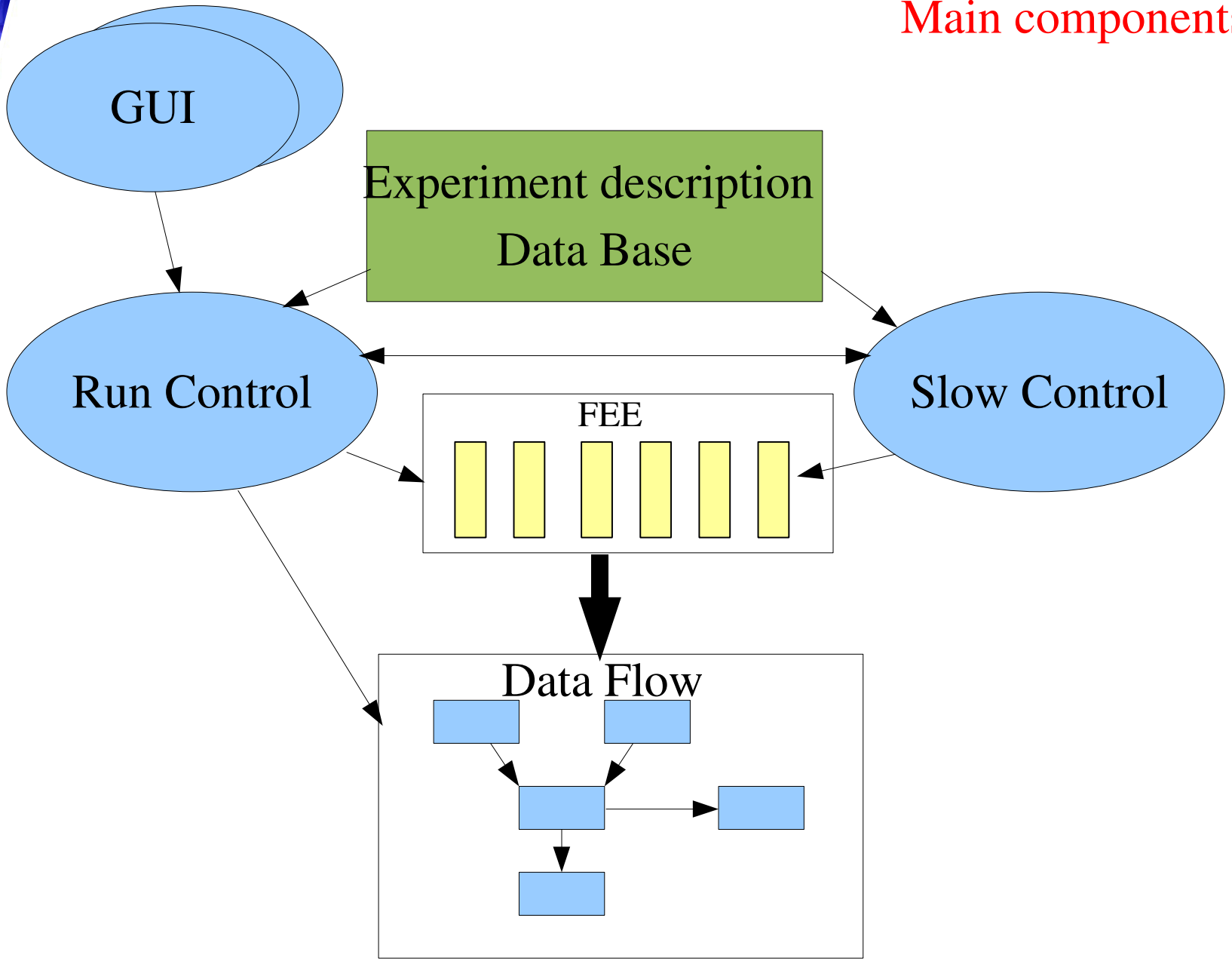
Software: - Event builder using timestamp
- Upgrade experiment control and setup

Future GANIL DAQ



- Requirements
  - small scale to large scale experiments
  - connect any frontend
  - trigger or triggerless systems
  - process time stamp data stream
  - highly modular acquisition system
  - run control for all the components
  - user friendly graphical interface

Main components



GANIL DAQ Development

- Some principles
  - Client/server architecture for Core of applications (eg Run control, Slow control)
  - Configurations saved in XML
  - Communications : SOAP messages
  - Errors : Log4j, Log4C++, ...

Slow Control

- **Main tasks**
  - Describe hardware configuration
  - Save / restore hardware configurations
  - Setup electronics
  - Monitor electronics
  - Handle errors and pass them to run control
  - Accept commands from outside (eg Run Control)
  - Several occurrences of GUI
- **Current GANIL slow control**
  - GUI well adapted to current configurations
  - To be redefined to separate GUI from kernel
  - Save format to be upgraded to use XML
  - Work to be evaluated

Run Control

- Main tasks
  - Configure DAQ for a run by selecting active components
  - Save/restore a configuration
  - Minimum set of commands to control all the components of the system (setup, start, stop...)
  - Monitor DAQ (status, data rate...)
  - Handle error messages
  - Log book
  - Scalable
- Needs a description of the architecture of the DAQ
- Development started

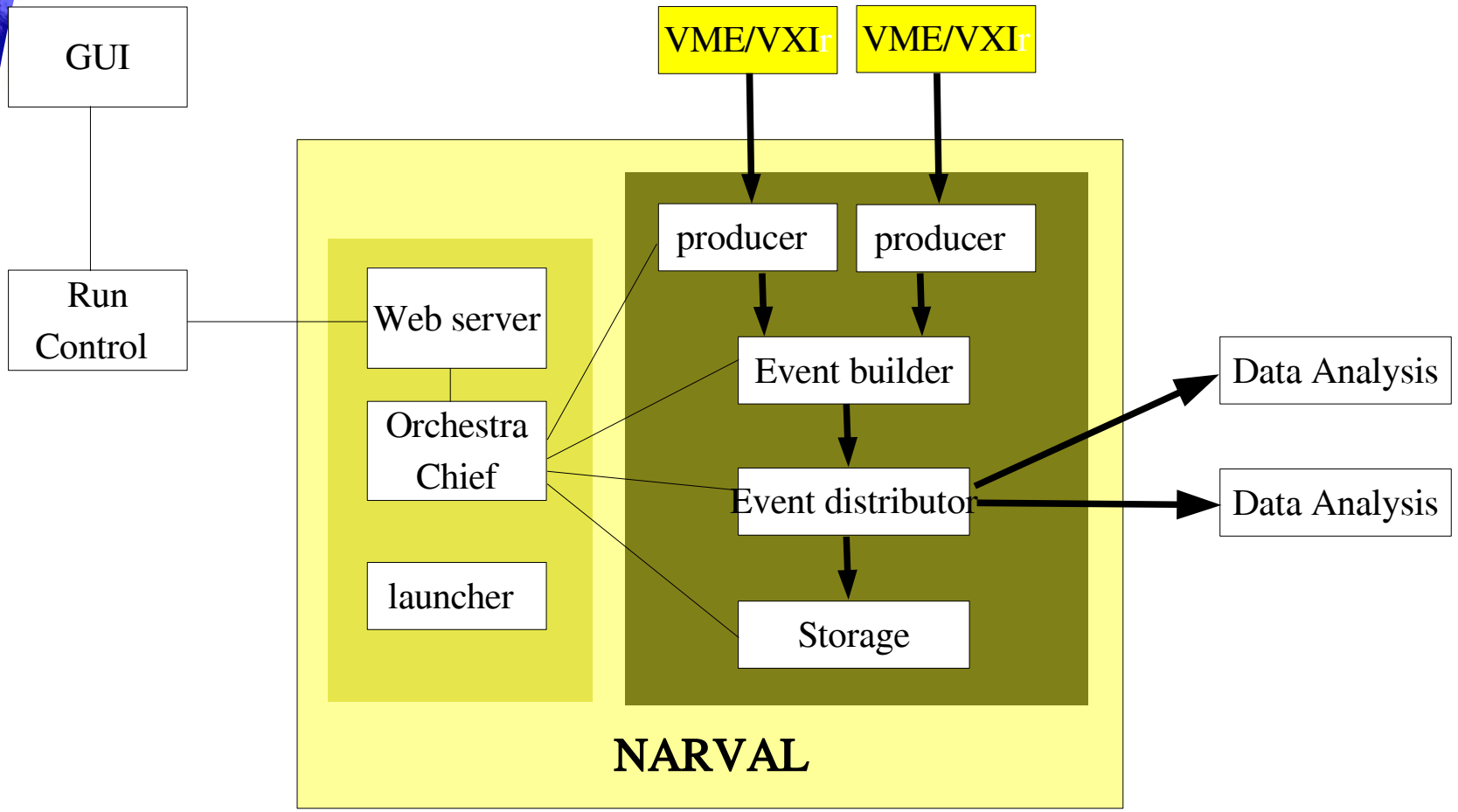
Data Flow

- **NARVAL**
  - Developped by IPN Orsay and CSNSM
  - Distributives Acquisition System
  - Developped in Ada95
    - OO programming
    - Strongly typed language
    - Robust applications
    - Distributed processes by using Annex E (CORBA equivalent)
  - Easy to link with C++
  - Used for AGATA DAQ
  - Collaboration with GANIL and LPC CAEN

NARVAL

- Main components
  - A main process to handle all configuration state (« chef d'orchestre »)
  - Set of actors to manage dataflow
    - Producer (get data from hardware)
    - Intermediary (act as a NxM soft switch that can filter data)
    - Consumer (end of data flow that can store or transform data)
  - Error handler (Log4Ada)
  - Launcher
- Data flow transport over Unix fifo, TCP/IP, Infiniband
- Communication via Web Services (SOAP) with the « chef d'orchestre »

Example of NARVAL in GANIL DAQ



Future GANIL DAQ

- General remarks
  - Graphical user interface must be intuive
  - Some tools are needed to diagnostic and solve problems at every level
  - Possibility to describe the system without having all the configuration on line
  - System must be evolutionary
  - It is difficult to create a new system without considering existing ones
  - Support to users is very important in laboratories like GANIL. Local team needs to have a very good knowledge of the systems to be supported.