Evolution of DAQ software at GANIL

Solutions adopted for Slow&Run Control

Frédéric Saillant

GANIL – Groupe Acquisition Physique

GANIL DAQ requirements

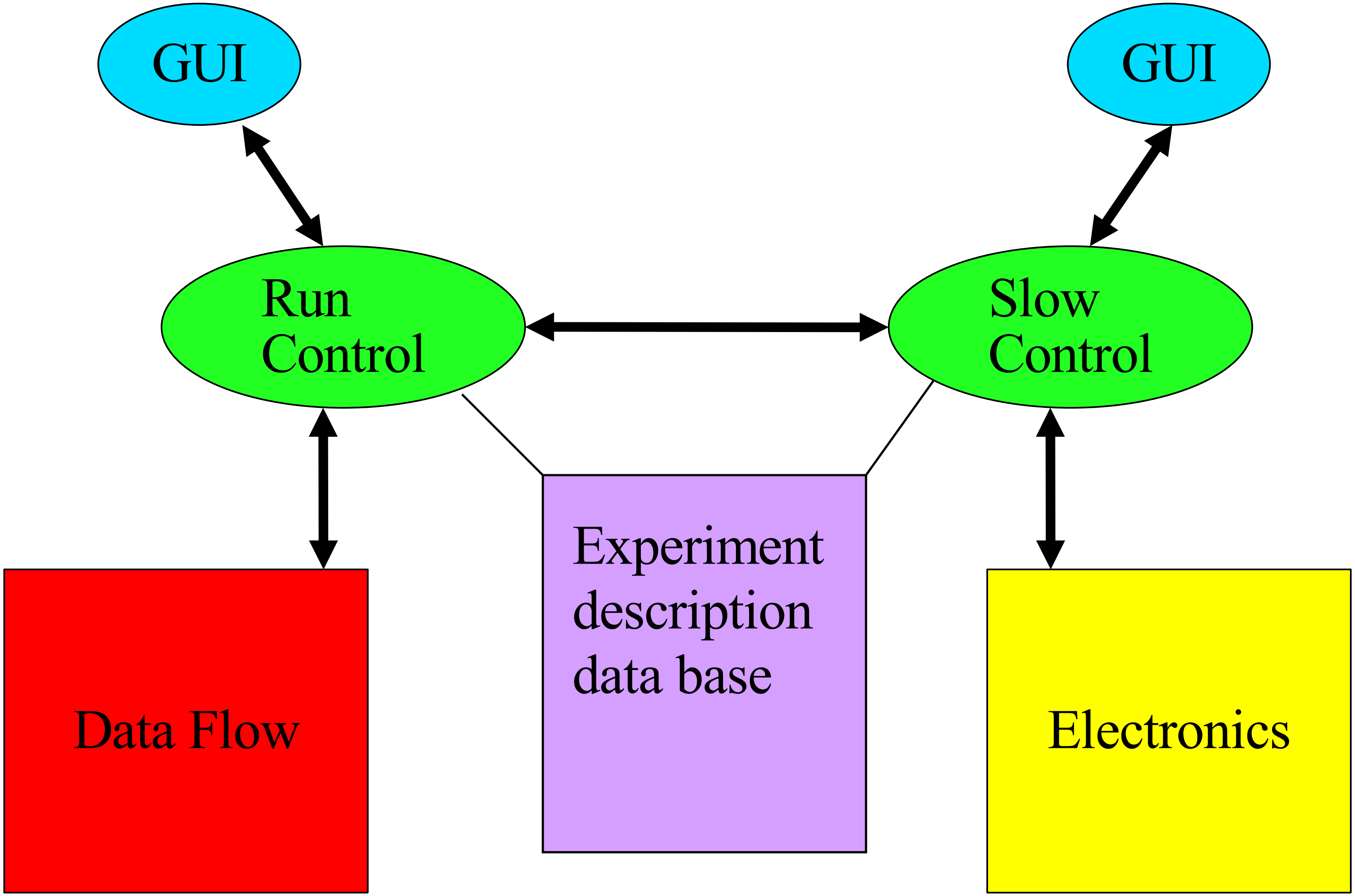
- small scale to large scale experiments
- connect any frontend
- trigger or triggerless systems
- process time stamp data streams
- highly modular acquisition system
- provide interfaces to plugin event filter algos
- run control and slow control for each component
- user-friendly graphical interfaces
- high data rates to be defined (> 100 MBytes/sec ?)

GANIL DAQ development

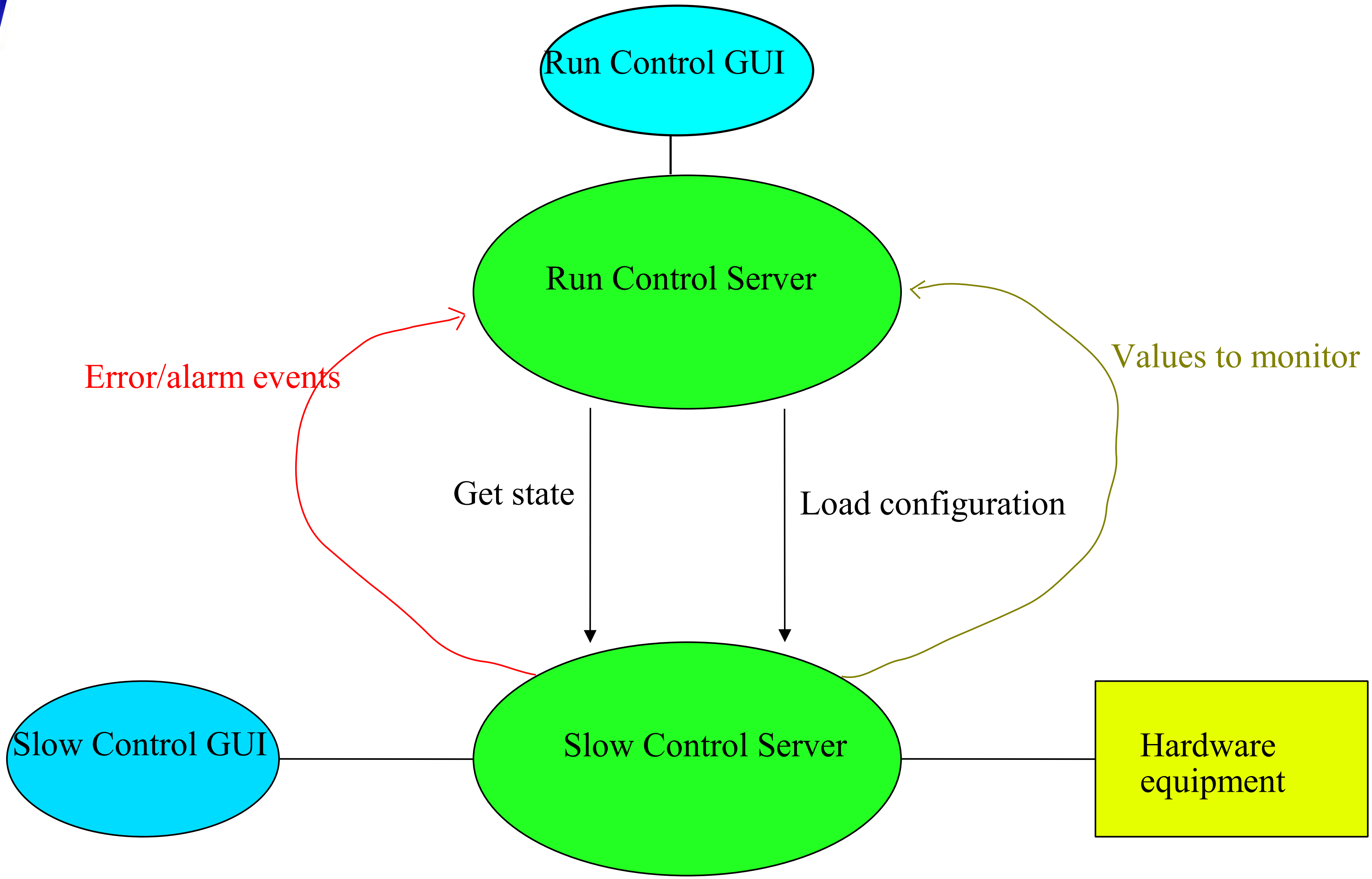
Some principles :

- Linux used on most nodes (even in FPGAs)
- Client/Server architecture for Core of applications (e.g. Run control, Slow control)
- Configurations saved in XML
- Communications : Web services (SOAP)
- Error/Info messages: Log4j, Log4C++, ...
- User-friendly graphical interfaces

Run Control and Slow Control



Interaction Run Control - Slow Control



Slow Control

Main tasks :

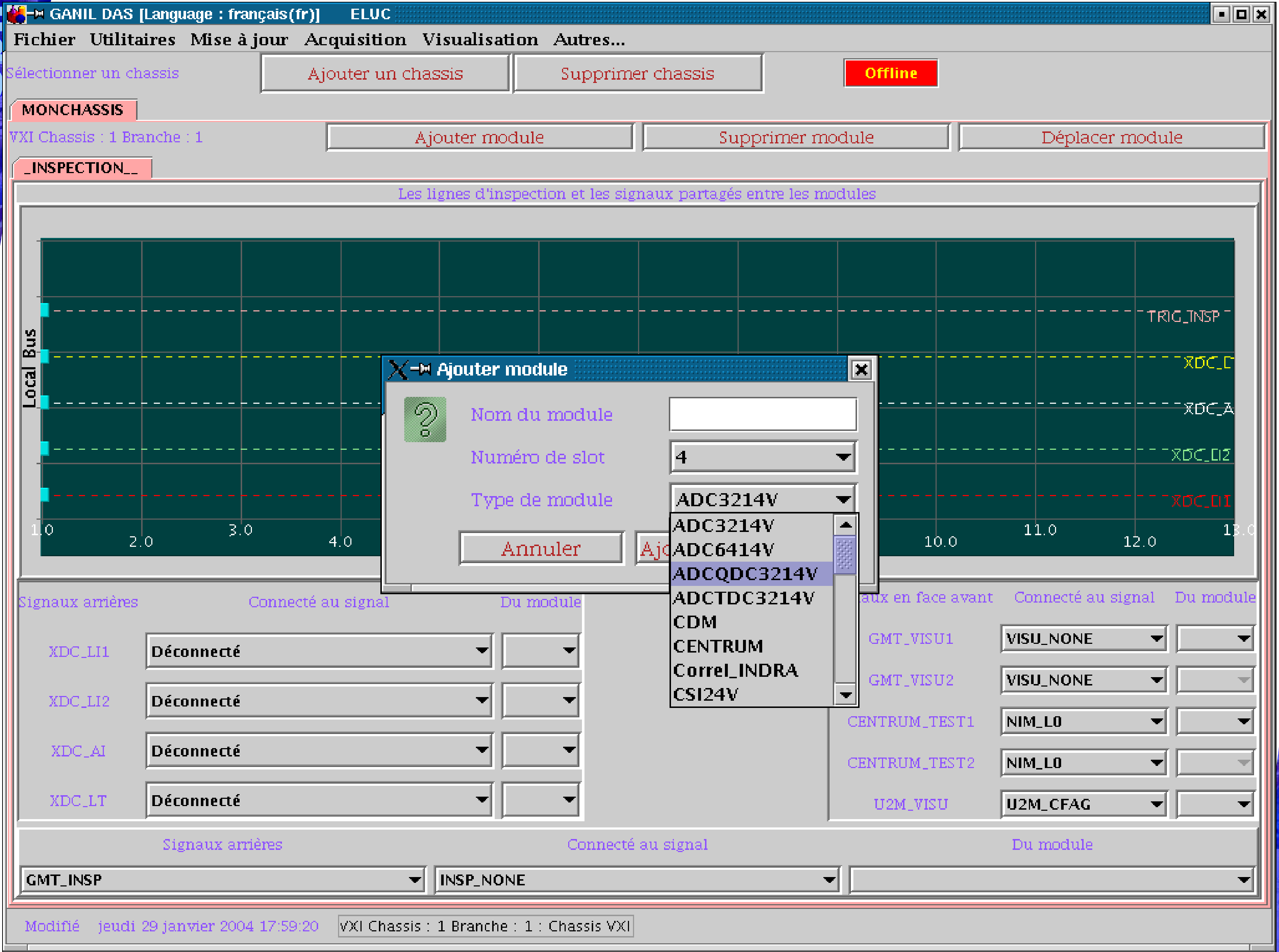
- Describe what hardware is to be controlled
- Save / restore hardware configurations
- Set-up hardware components (write registers)
- Monitor hardware components (e.g. temperature)
- Handle error/alarm events and pass them to Run Control
- Accept commands from outside (e.g. Run Control)
- Core of the application separated from GUI
- Several occurrences of GUI

Slow Control

Current GANIL Slow Control : DAS setup panel

- GUI well adapted to current configurations
  - Describe hardware configuration
  - Save / restore hardware configurations
  - Set-up electronics
  - Monitor electronics for specific cards (MUVI)
- To be reworked to
  - Separate GUI from the core of the application
  - Handle errors and pass them to Run Control
  - Accept commands from outside (e.g. Run Control)
  - Several occurrences of GUI
- Storage format to be upgraded to use XML
- Work to be evaluated

DAS setup panel



VXI Chassis : 1 Branche : 1

Les lignes d'inspection et les signaux partagés entre les modules

Nom du module

Numéro de slot: 4

Type de module: ADC3214V

ADC3214V
ADC6414V
ADCQDC3214V
ADCTDC3214V
CDM
CENTRUM
Correl_INDRA
CSI24V

Signaux arrières	Connecté au signal	Du module
XDC_LI1	Déconnecté	
XDC_LI2	Déconnecté	
XDC_AI	Déconnecté	
XDC_LT	Déconnecté	

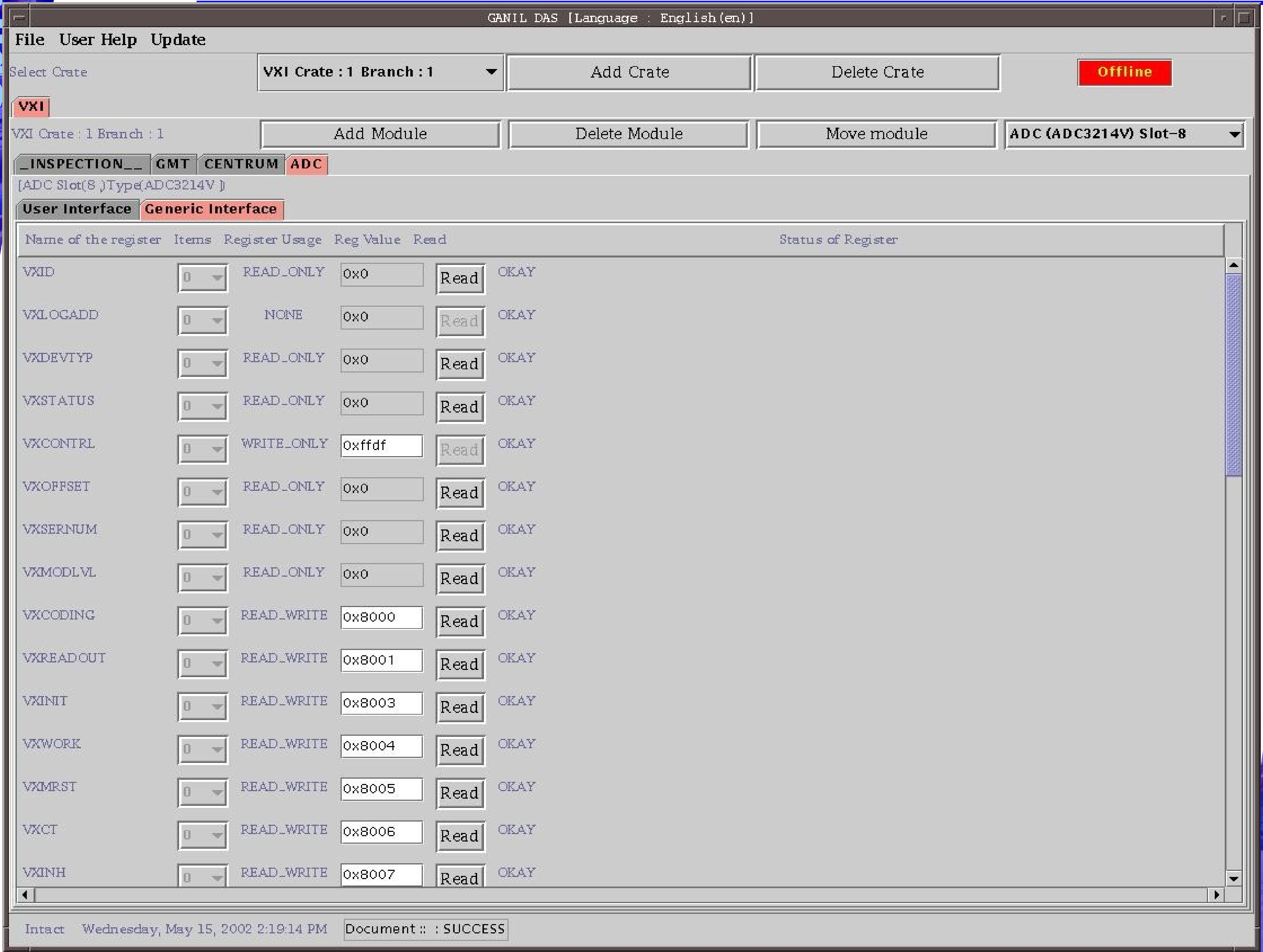
Signaux en face avant	Connecté au signal	Du module
GMT_VISU1	VISU_NONE	
GMT_VISU2	VISU_NONE	
CENTRUM_TEST1	NIM_L0	
CENTRUM_TEST2	NIM_L0	
U2M_VISU	U2M_CFAG	

Signaux arrières	Connecté au signal	Du module
GMT_INSP	INSP_NONE	

Modifié jeudi 29 janvier 2004 17:59:20

VXI Chassis : 1 Branche : 1 : Chassis VXI

DAS setup panel



GANIL DAS [Language : English(en)]

Select Crate: VXI Crate : 1 Branch : 1

VXI Crate : 1 Branch : 1

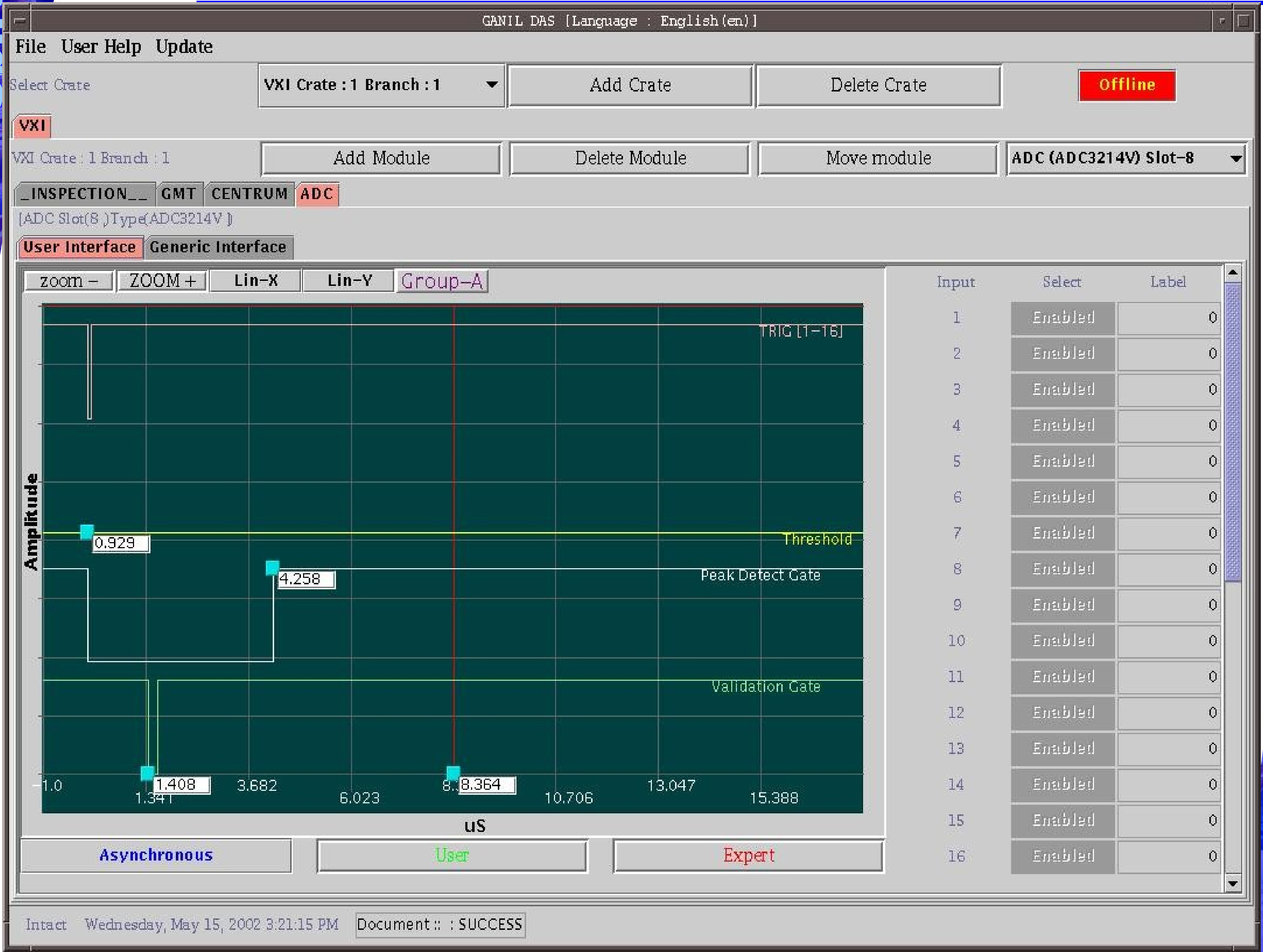
ADC (ADC3214V) Slot-8

[ADC Slot(8)Type(ADC3214V)]

Name of the register	Items	Register Usage	Reg Value	Read	Status of Register
VXID	0	READ_ONLY	0x0	Read	OKAY
VXLOGADD	0	NONE	0x0	Read	OKAY
VXDEVTYP	0	READ_ONLY	0x0	Read	OKAY
VXSTATUS	0	READ_ONLY	0x0	Read	OKAY
VXCONTRL	0	WRITE_ONLY	0xffdf	Read	OKAY
VXOFFSET	0	READ_ONLY	0x0	Read	OKAY
VXSERNUM	0	READ_ONLY	0x0	Read	OKAY
VXMODLVL	0	READ_ONLY	0x0	Read	OKAY
VXCODING	0	READ_WRITE	0x8000	Read	OKAY
VXREADOUT	0	READ_WRITE	0x8001	Read	OKAY
VXINIT	0	READ_WRITE	0x8003	Read	OKAY
VXWORK	0	READ_WRITE	0x8004	Read	OKAY
VXMRST	0	READ_WRITE	0x8005	Read	OKAY
VXCT	0	READ_WRITE	0x8006	Read	OKAY
VXINH	0	READ_WRITE	0x8007	Read	OKAY

Intact Wednesday, May 15, 2002 2:19:14 PM Document :: : SUCCESS

DAS setup panel



VXI Crate : 1 Branch : 1

VXI Crate : 1 Branch : 1

ADC (ADC3214V) Slot-8

[ADC Slot(8)Type(ADC3214V)]

Input	Select	Label
1	Enabled	0
2	Enabled	0
3	Enabled	0
4	Enabled	0
5	Enabled	0
6	Enabled	0
7	Enabled	0
8	Enabled	0
9	Enabled	0
10	Enabled	0
11	Enabled	0
12	Enabled	0
13	Enabled	0
14	Enabled	0
15	Enabled	0
16	Enabled	0

Intact Wednesday, May 15, 2002 3:21:15 PM Document :: : SUCCESS

Run Control

Main tasks :

- Configure DAQ for a run by selecting active components
- Save/restore a configuration
- Commands to control all the active components of the system (setup, start, stop...)
- Monitor DAQ (status, data rates...)
- Handle error/info messages
- Log book
- User-friendly graphical interface, separated from the core of the application

Run Control

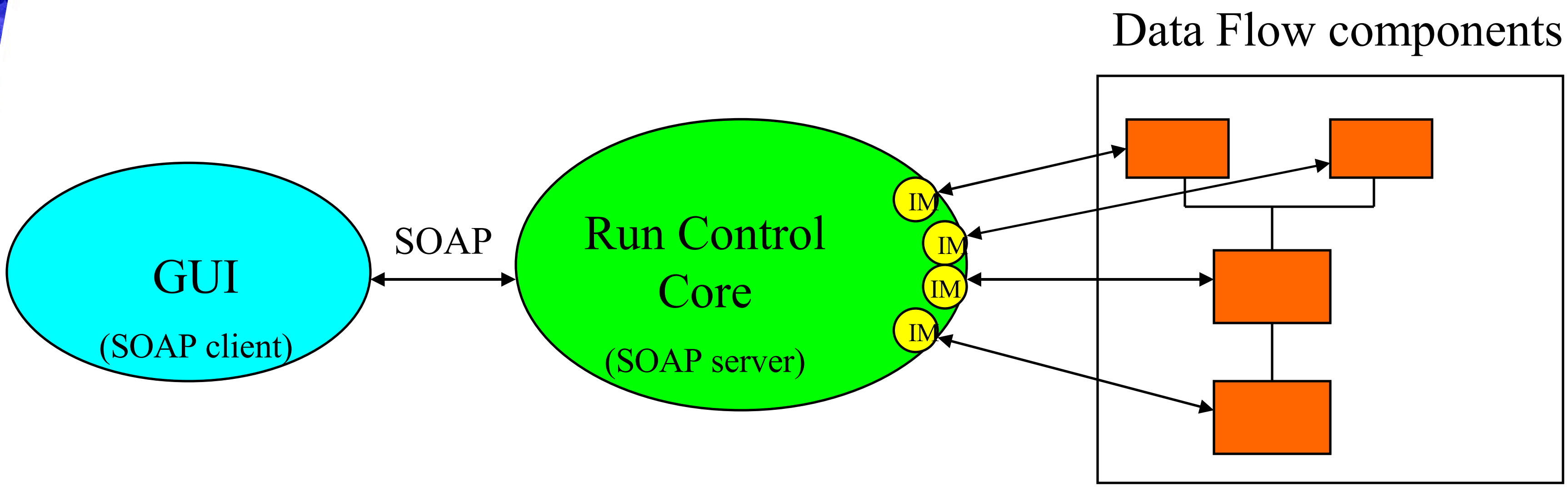
Current GANIL Run Control : DAS command panel

- **Available basic functionnalities :**
  - Configure DAQ for a run by selecting active components
  - Save/restore a configuration
  - Minimum set of commands to control all the active components of the system (setup, start, stop...)
  - Monitor DAQ (status, data rates...)
  - Scalable for simple systems

- **Missing functionnalities :**
  - Core of application separated from the GUI
  - Handle error/info messages
  - Log book
  - Scalabality for complex experiments

New Development started

Run Control : new architecture



- Run Control Core accesses data flow components with specific communication protocols encapsulated in « Instrument Managers »
- Run Control Core written in C++ with gSOAP library
- WSDL file generated by gSOAP
- Java GUI integrates SOAP client stub thanks to WSDL file

DAQ Data Flow

NARVAL

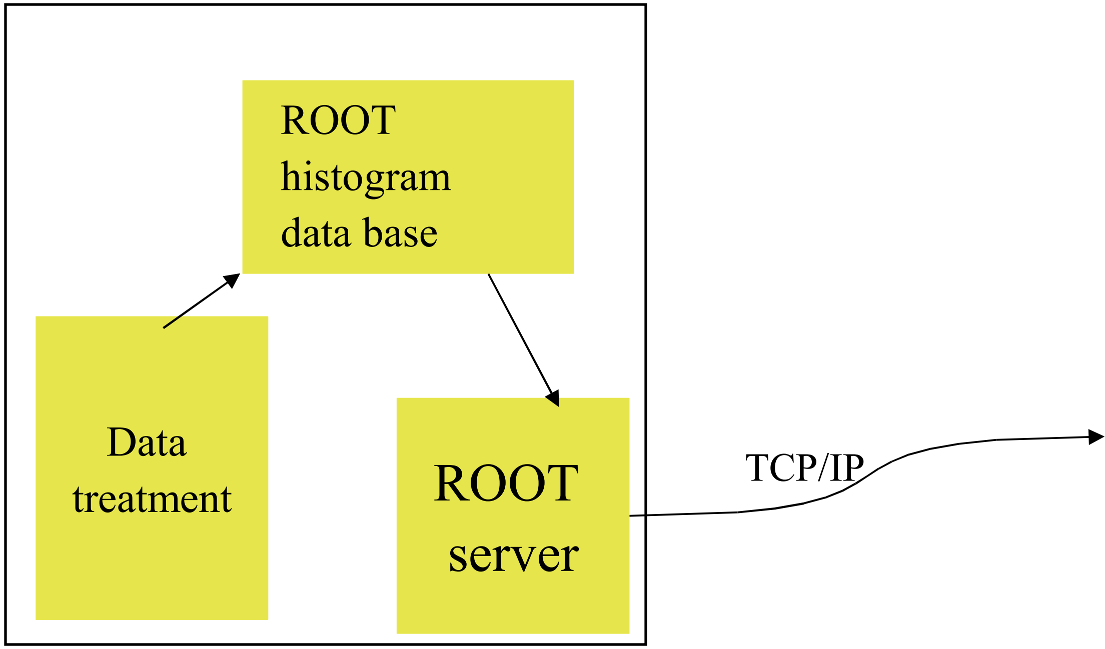
- Originally developped by IPN Orsay
- Today, collaborative development with IPNO, CSNSM, GANIL, LPC Caen
- Distributed Acquisition System
- Developped in Ada95
  - Object Oriented programming
  - Strongly typed language
  - Robust applications
  - Distributed processes by using Annex E (CORBA equivalent)
- Easy to link with C++
- Used for AGATA DAQ
- *Web site : http://narval.in2p3.fr/*

NARVAL

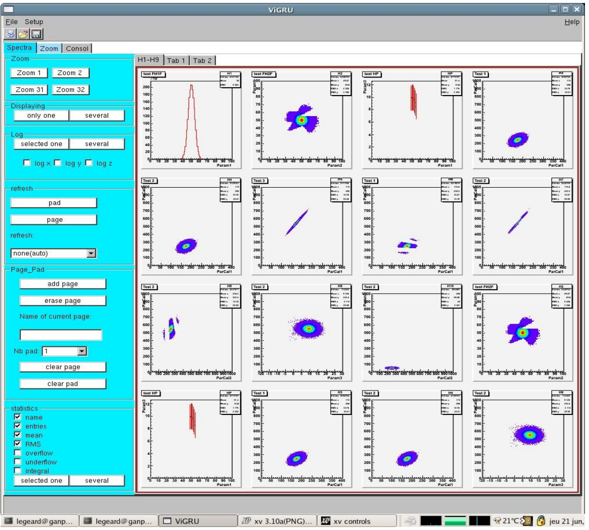
- **Main components :**
  - A main process to handle the state of all the configuration (Coordinator)
  - Set of actors to manage the data flow
    - Producer : input of data flow (hardware or other DAQ)
    - Intermediary : acts as a NxM soft switch that can filter data
    - Consumer : end of data flow (data storage, output to other DAQ,...)
  - Logging of error/info messages (Log4Ada)
- Data flow transport over Unix fifo, TCP/IP, Infiniband
- Communication via Web Services (SOAP) with the « Coordinator »

ROOT in NARVAL

GANIL ROOT Utilities



NARVAL Actor



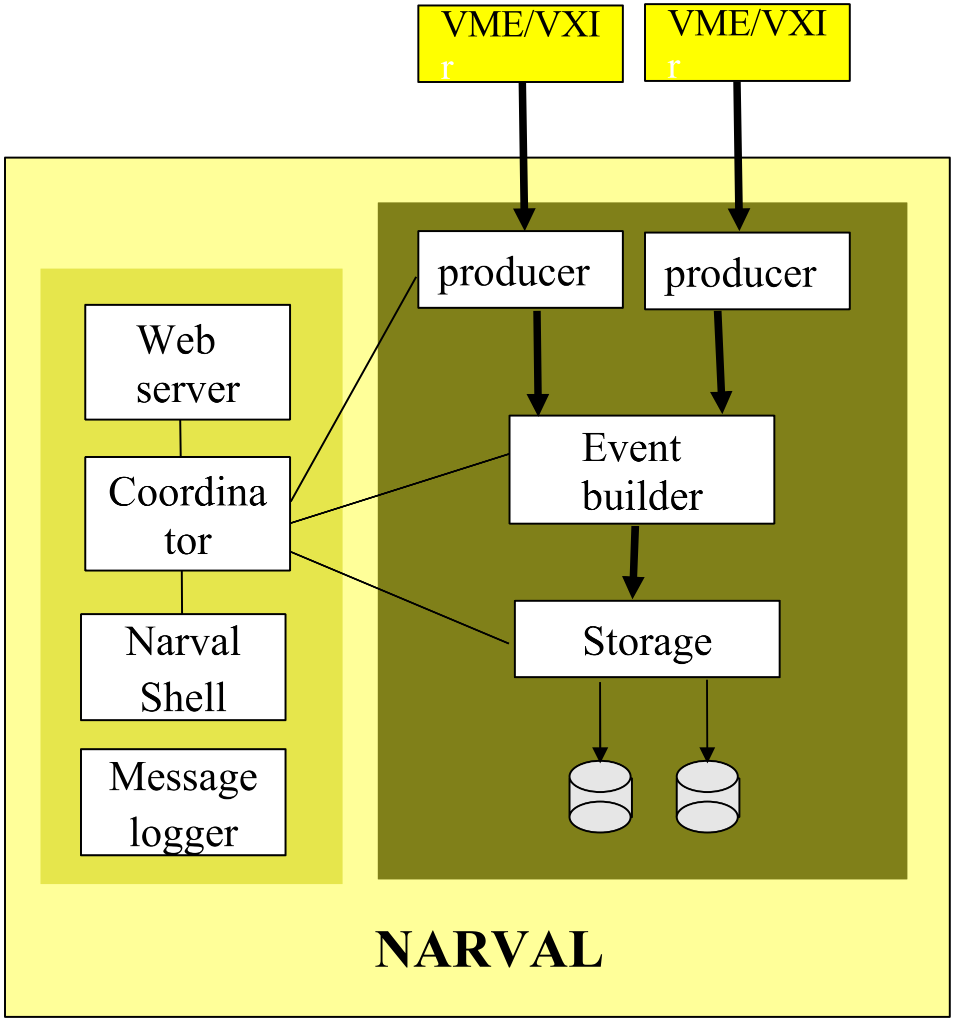
ViGRU

ROOT Spectra display

- ✓ ADA binding for basic commands to create, destroy, increment histograms
- ✓ ROOT library directly accessible for C++ programmers

NARVAL in GANIL DAQ

Test configuration



Titre

Sous-titre

Auteur

GANIL-Caen

- Introduction
    - Premier point
- Chapitre 1
    - Section 1
- …