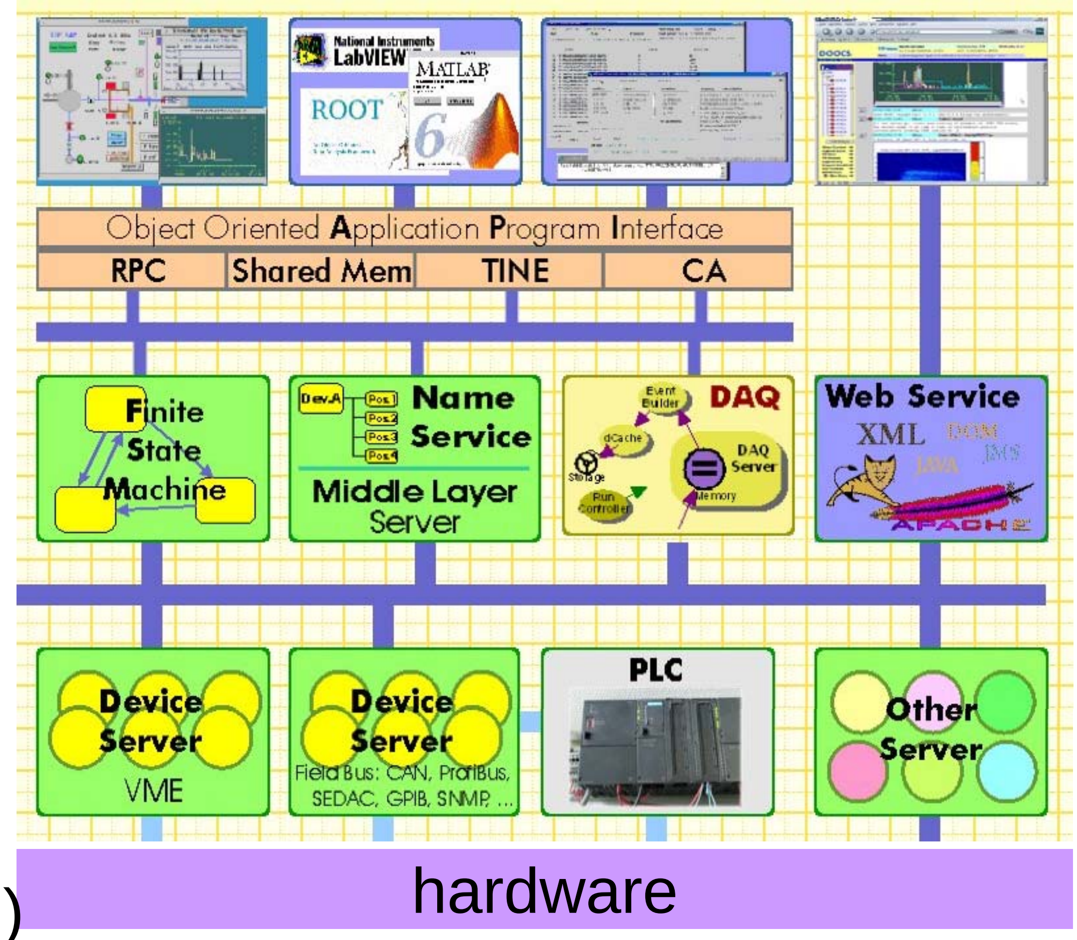
DOOCS framework for CALICE DAQ software

Valeria Bartsch,
UCL

Tao Wu
RHUL

DOOCS overview

- 3 layers
- common APIs
- modular design
- multi protocol (RPC, TINE, EPICS, shared memory)
- device level (~200 server types)
- middle layer (FSM, FB, DAQ)



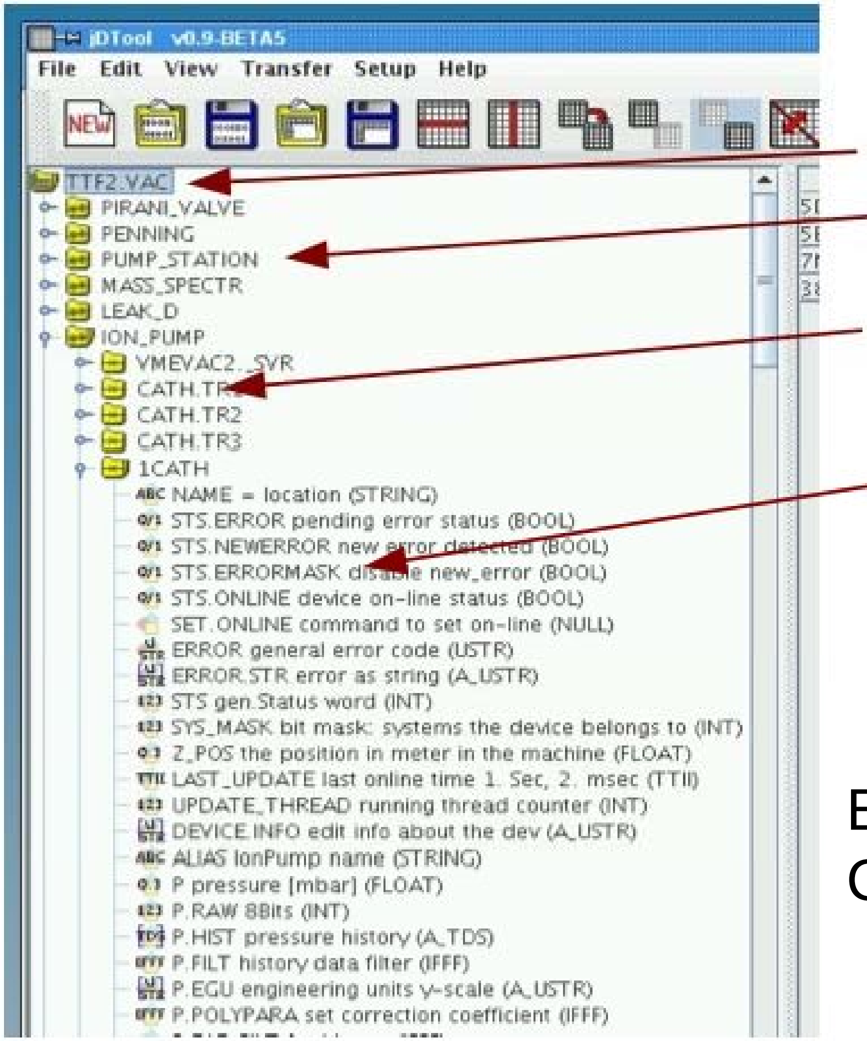
User Interface

Program Interface

Middle Layer

Hardware interface

ENS naming service



```
TTF2.VAC
  PIRANI_VALVE
  PENNING
  PUMP_STATION
  MASS_SPECTR
  LEAK_D
  ION_PUMP
    VMEVAC2._SVR
    CATH.TR1
    CATH.TR2
    CATH.TR3
    1CATH
      ABC NAME = location (STRING)
      0/1 STS.ERROR pending error status (BOOL)
      0/1 STS.NEWERROR new error detected (BOOL)
      0/1 STS.ERRORMASK disable new_error (BOOL)
      0/1 STS.ONLINE device on-line status (BOOL)
      SET.ONLINE command to set on-line (NULL)
      ERROR general error code (USTR)
      ERROR.STR error as string (A_USTR)
      123 STS gen.Status word (INT)
      123 SYS_MASK bit mask: systems the device belongs to (INT)
      0.1 Z_POS the position in meter in the machine (FLOAT)
      TTII LAST_UPDATE last online time 1. Sec, 2. msec (TTII)
      123 UPDATE_THREAD running thread counter (INT)
      DEVICE.INFO edit info about the dev (A_USTR)
      ABC ALIAS IonPump name (STRING)
      0.1 P pressure [mbar] (FLOAT)
      123 P.RAW 8Bits (INT)
      TDS P.HIST pressure history (A_TDS)
      IFFF P.FILT history data filter (IFFF)
      P.EGU engineering units y-scale (A_USTR)
      IFFF P.POLYPARA set correction coefficient (IFFF)
```

FACILITY == Accelerator

DEVICE == Type of a device

LOCATION == position inside the accelerator

PROPERTY == list of properties

Example:
CALICE.ECAL/ODR/ODR1/STATUS

ENS naming service: proposal for CALICE

Our proposal for the naming service:

FACILITY:	CALICE.ECAL, CALICE.AHCAL, CALICE.DHCAL
DEVICE:	ODR, LDA, DIF, ASIC1, ASIC2, ASIC3
LOCATION:	ODR1, ODR2, ODRX LDA1, LDA2, LDAX DIF1, DIF2, DIFX
PROPERTY:	????

⇒ need to get input from hardware colleagues about properties of the devices

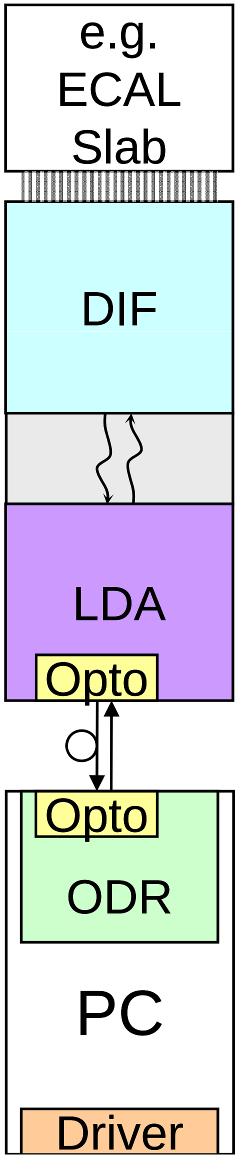
ENS naming service: hierachical DAQ system

- send data to DIF by wrapper through ODR and LDA (have switch to configure debugging modes which go directly to the LDA or DIF)
- ENS naming service can signal connections by additional properties, e.g. for device DIF:

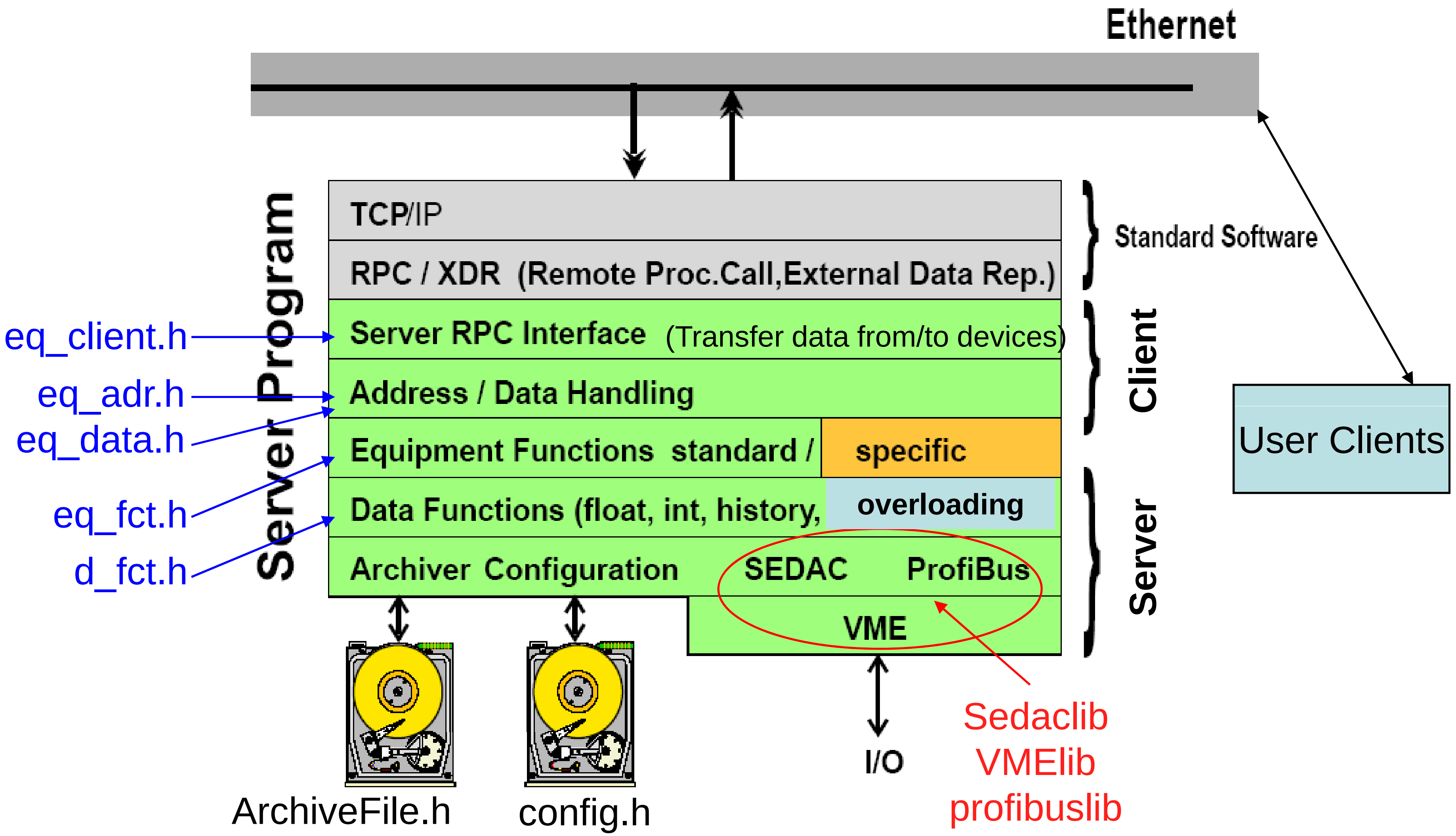
CALICE.ECAL/DIF/DIF1/ODR_CON

CALICE.ECAL/DIF/DIF1/LDA_CON

CALICE.ECAL/DIF/DIF1/DEBUG_MODE



Device Servers



Device Servers for CALICE

• servers will update/poll information every few seconds (property can be set in configuration)
⇒ can be used for monitoring & data taking

▪ data functions can be overloaded to set registers in hardware and to get registers
⇒ can be used for configuration & getting values which are typically not monitored (do not use it to read out too many values, because it directly accesses the memory)

• interfaces to hardware need to be communicated to and discussed with DAQ software group

Possible functions for ODR

```
class ODRCard  {
    ODRCard();
    virtual ~ODRCard();
    virtual void InitODR ();
    virtual void MappingODR();
    virtual void UnMappingODR();
    virtual void ReadODR();
    virtual void WriteODR();
    virtual void ODRIOCTL();
    virtual void ODRStatus();
    virtual void ODRCMD();
    virtual void ODRMonitor();
    virtual void ODRSetLocalVal();
    virtual void ODRReadLocalVal();
    virtual void GetLiveList();
}
// with arguments for each…
```

open (“/dev/odr3”, O_RDWR);

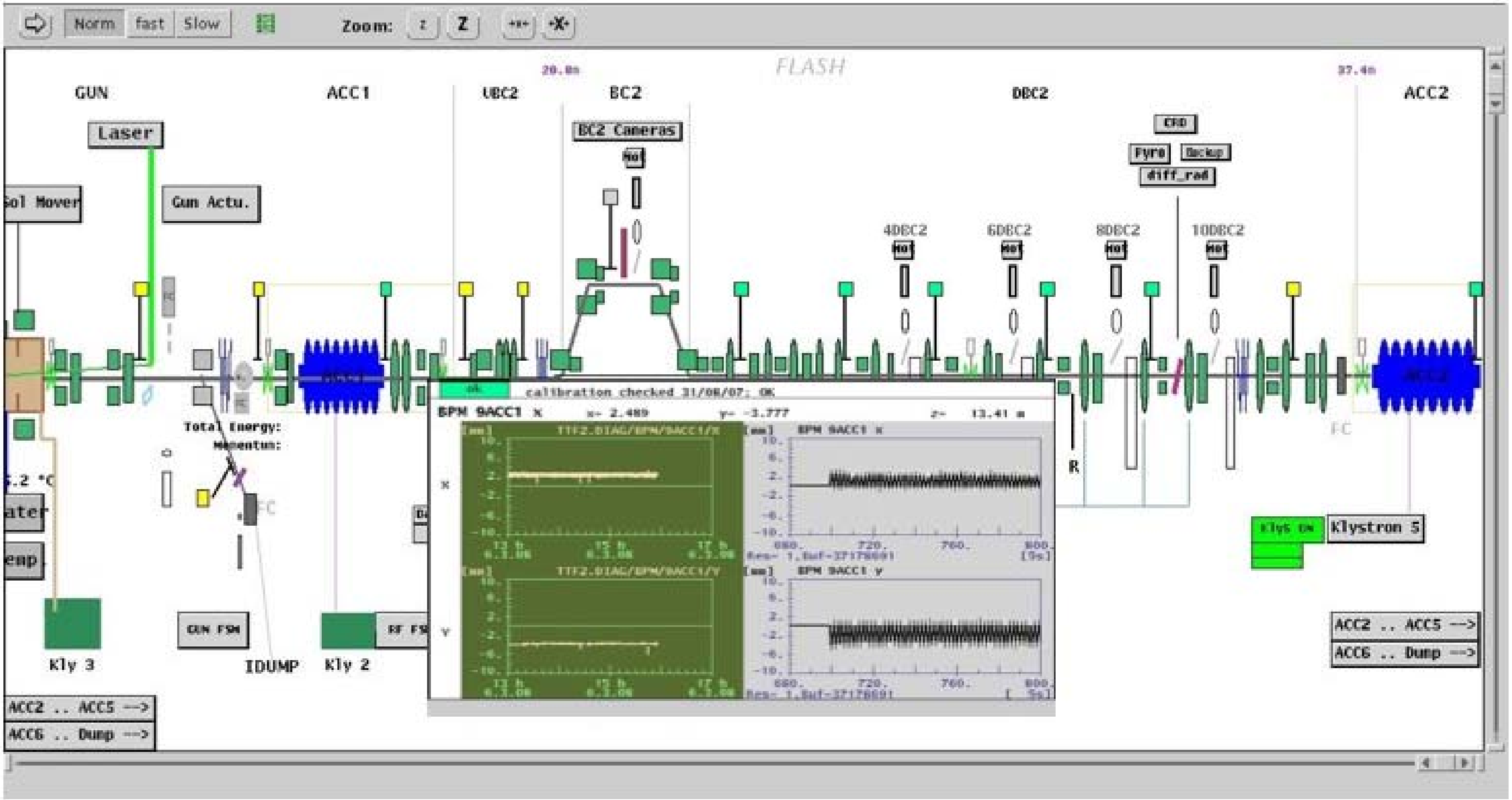
Initialize some parameters, and Mapping ODR device involved, ioctl(… , INIT); etc.

Possible Properties for ODR

```
// Intrinsic Properties
Channels 1010 // Channel activation/selection
Mode Send // Send or Receive mode
ExtractWord ID // Selected Word can be extracted from data
Debug 0 // 0 or 1 for debugging
DataGen 0 // 0 or 1 for external data generator
DataBuf 10 // number of data buffers
EventGroup ??? // grouping of events
```

Example of monitoring GUI

DOOCS Data Display (DDD)



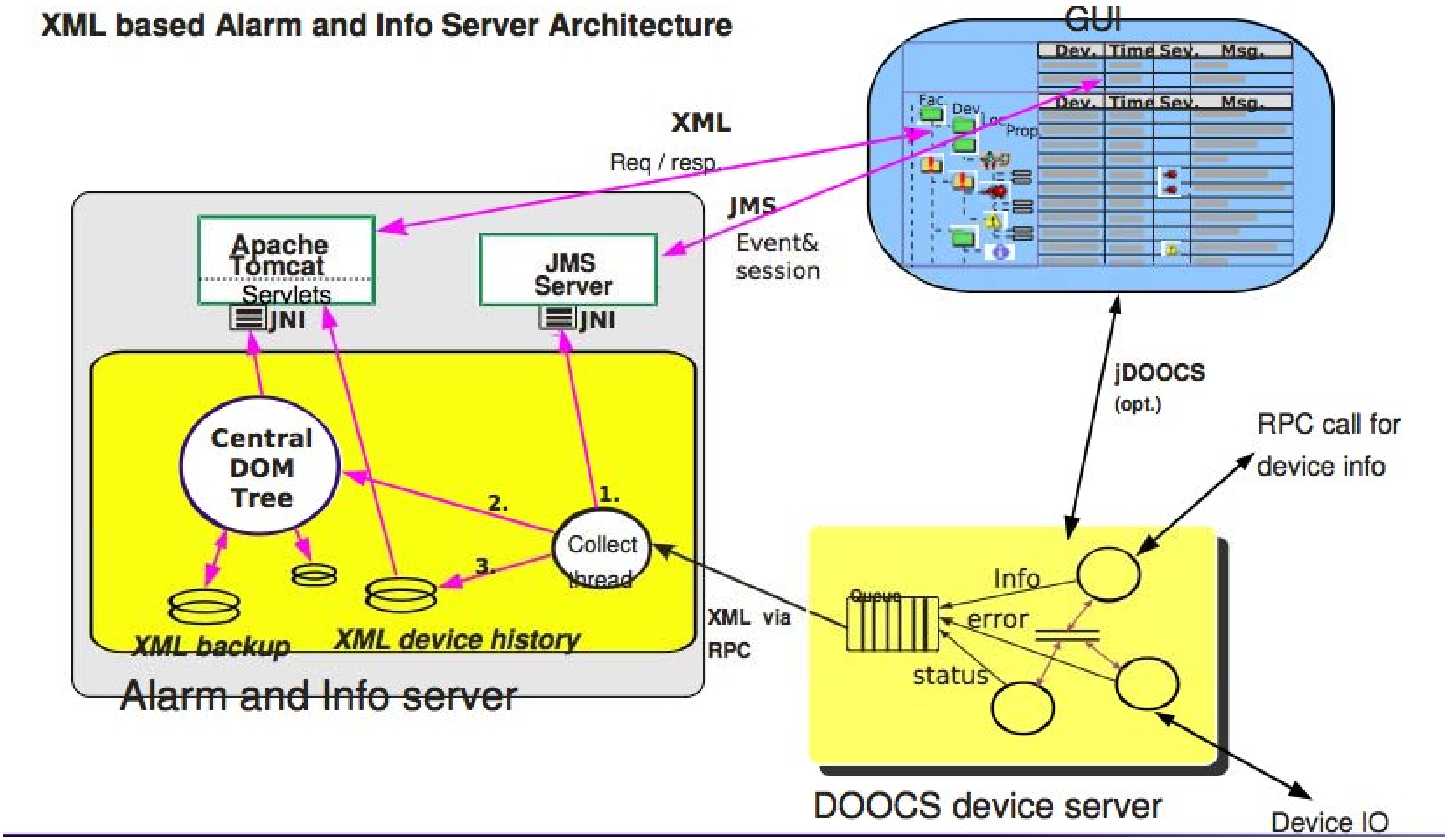
calibration checked 31/08/07: OK

BPM 9ACC1 x x= 2.489 y= -3.777 z= 13.41 m

Example of monitoring GUI

- monitoring can be drawn by special program
- easy to use, can be even handed to shifters
- many nice features:
    - click to get to histos,
    - display of broken links, etc.
- for CALICE application nothing done yet, will be added at a later stage

Alarm handling

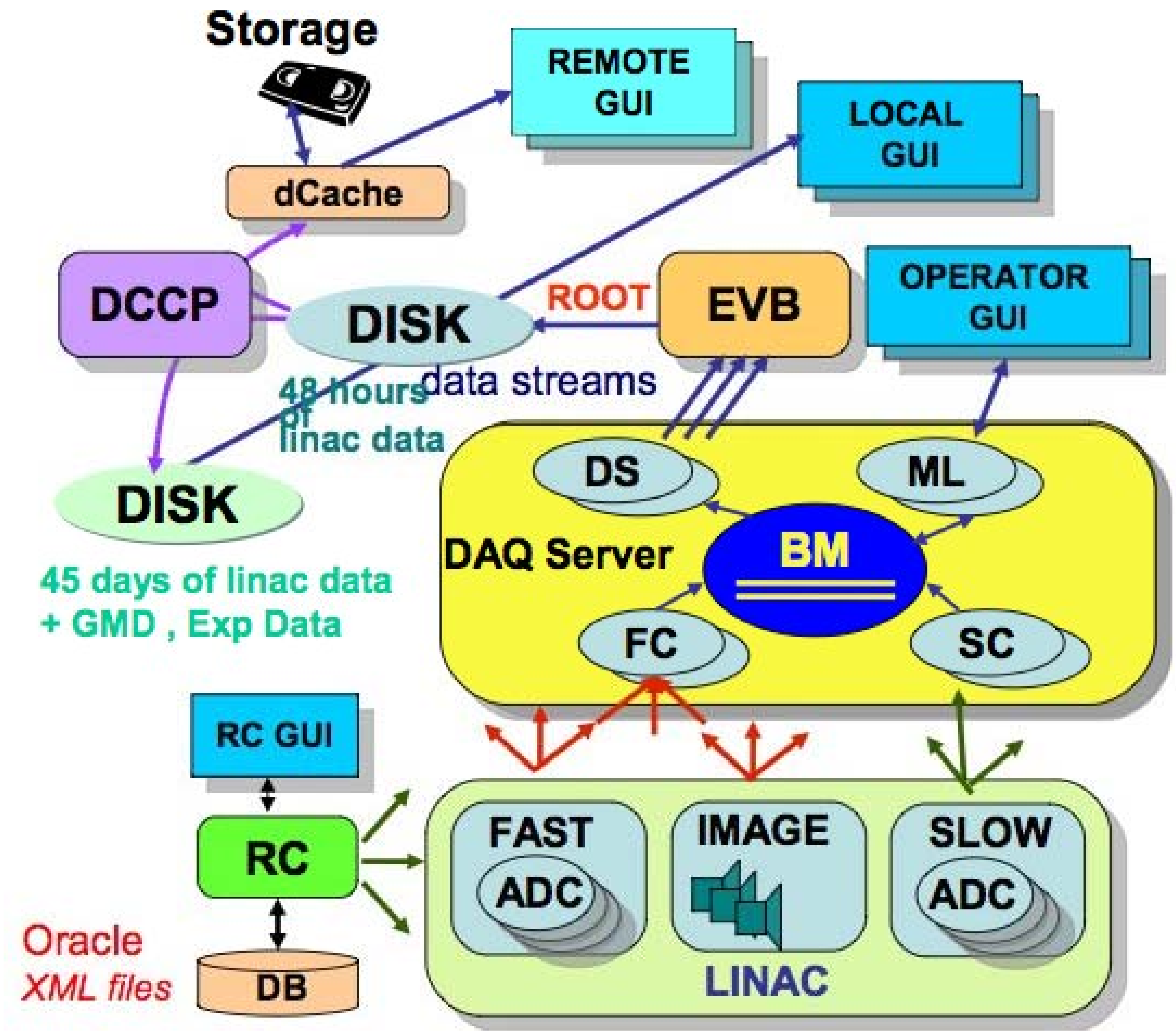


XML based Alarm and Info Server Architecture

Alarm handling - CALICE

- implementation of alarm handling within CALICE application will be added at a more mature state of the software
- however configuration considers alarm handling already at this stage

DAQ software



FC/SC:
Fast/Slow
Collector

BM:
Buffer Manager

EVB:
Event Builder

Example with dummy data has been successfully tested

Conclusion & Outlook

• how to apply DOOCS for the CALICE technical prototype is well understood

• the basic design for the CALICE application is ready

• interface to the hardware is the starting point of the implementation of the CALICE project within DOOCS

⇒ next step is to get feedback from the hardware guys to build the hardware interfaces and settle on the naming conventions