

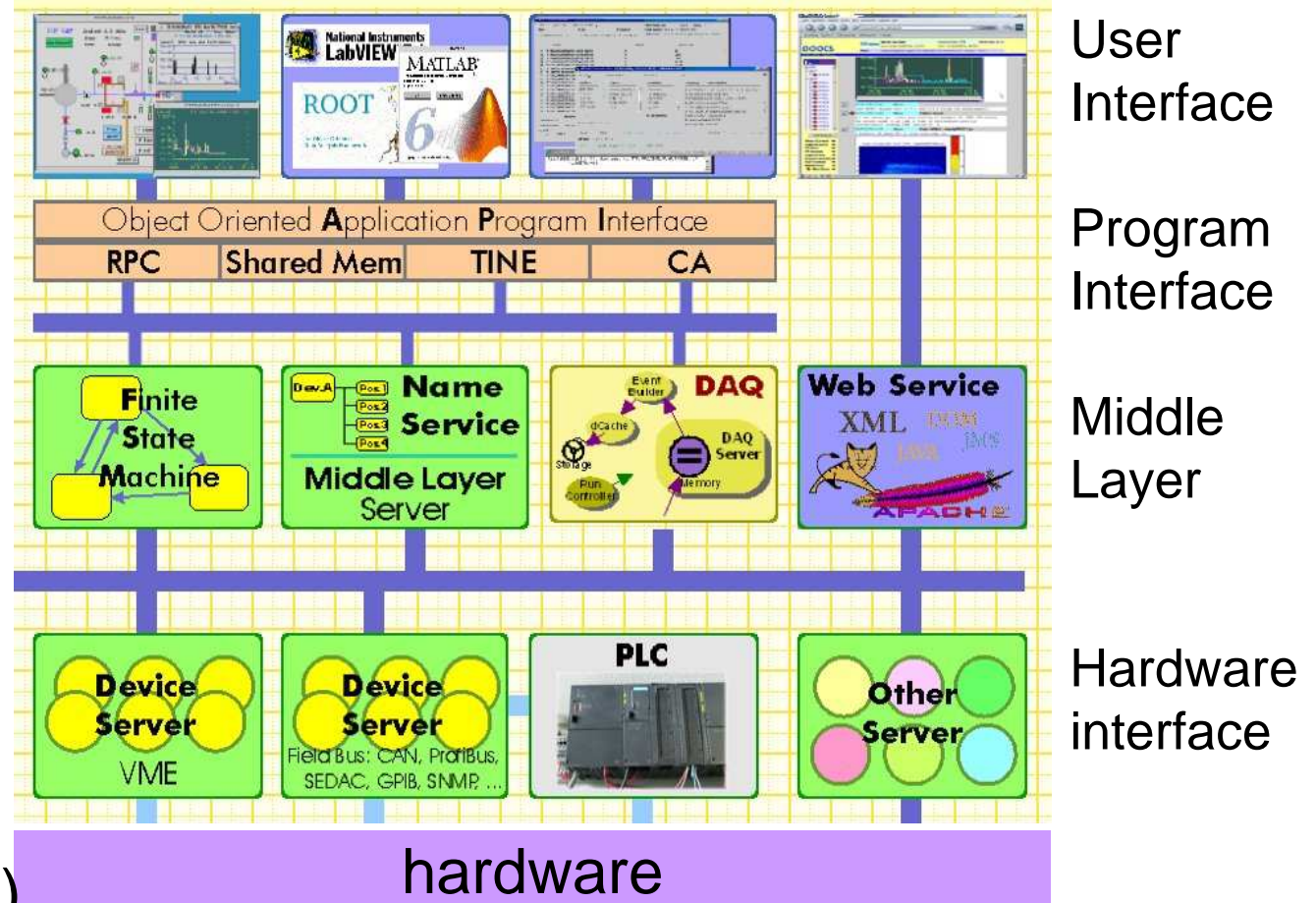
DOOCS framework for CALICE DAQ software

Valeria Bartsch,
UCL

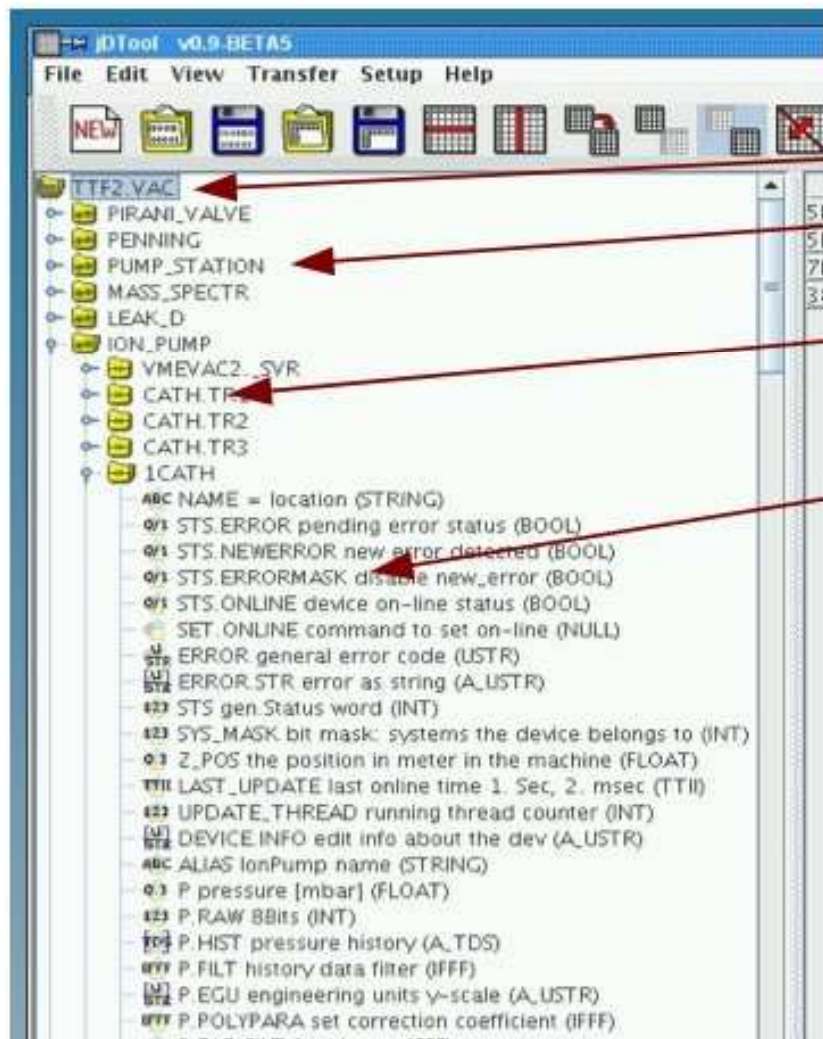
Tao Wu
RHUL

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)



ENS naming service



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

LOCATION: ODR1, ODR2, ODRX
LDA1, LDA2, LDAX
DIF1, DIF2, DIFX

PROPERTY: ????

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

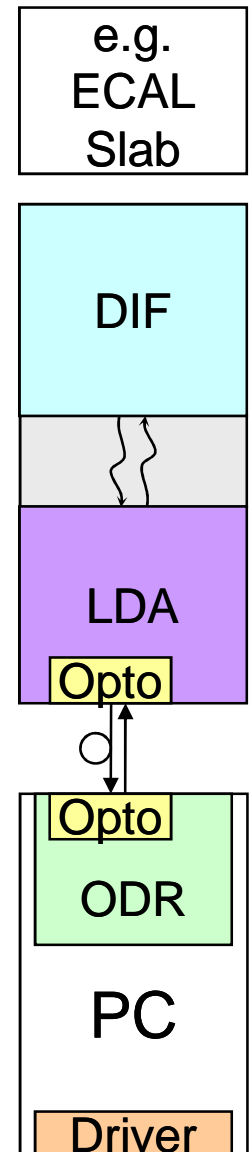
ENS naming service: hierachical DAQ system

- 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:

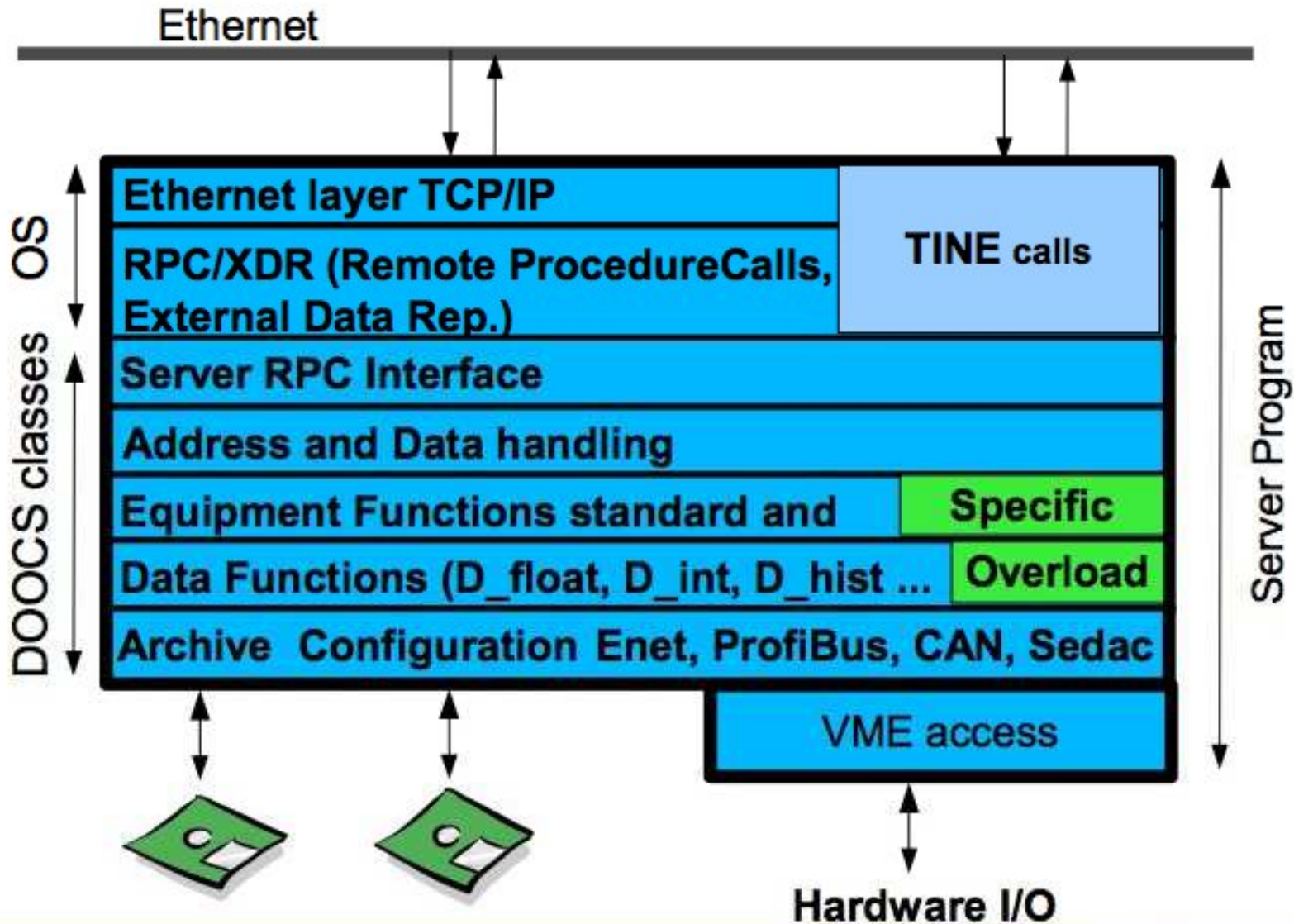
CALICE.ECAL/DIF/DIF1/ODR_CON

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

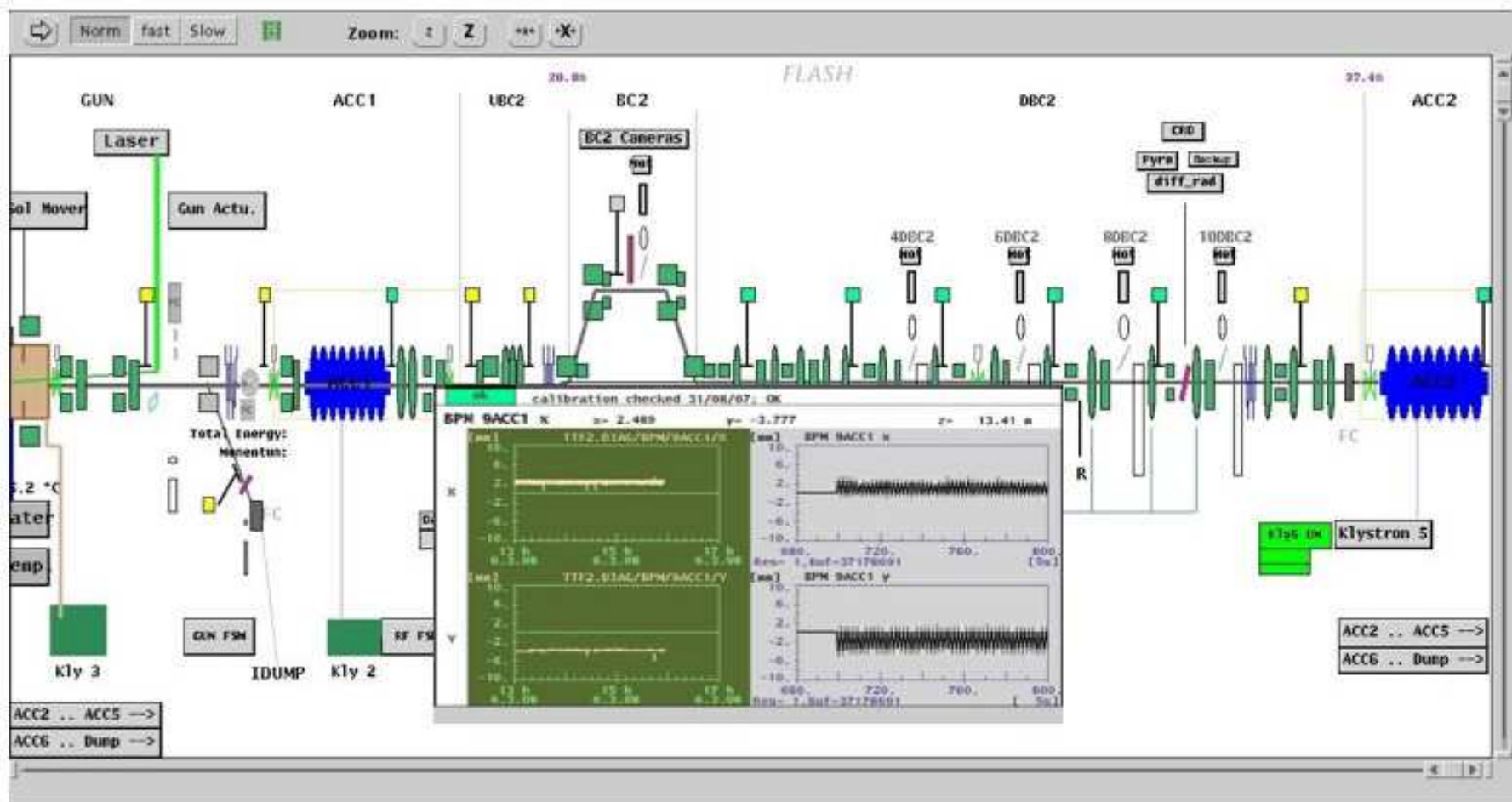
§ data functions can be overloaded to set registers in hardware and to get registers

⇒ 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

Example of monitoring GUI

DOOCS Data Display (DDD)

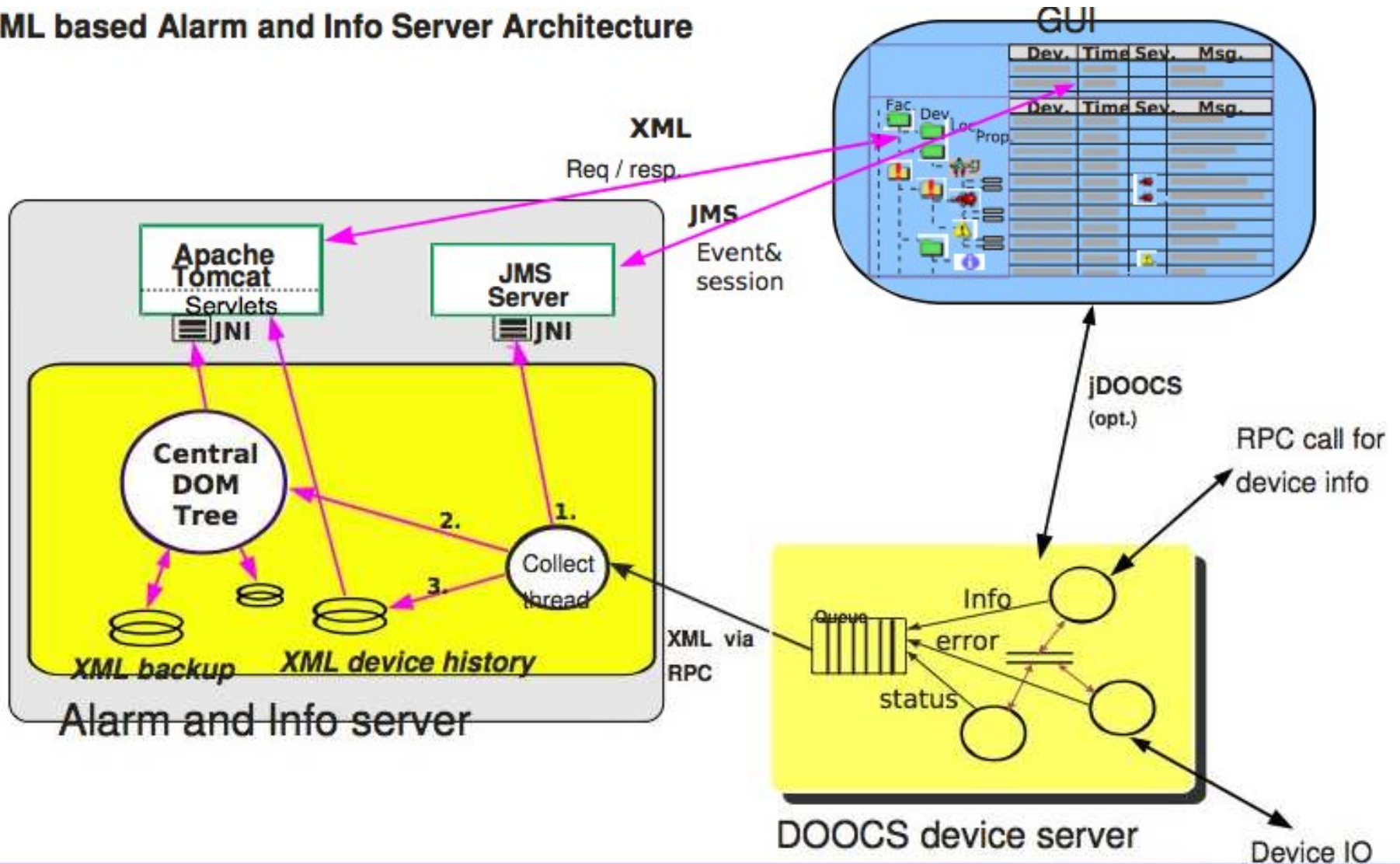


Example of monitoring GUI

- 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

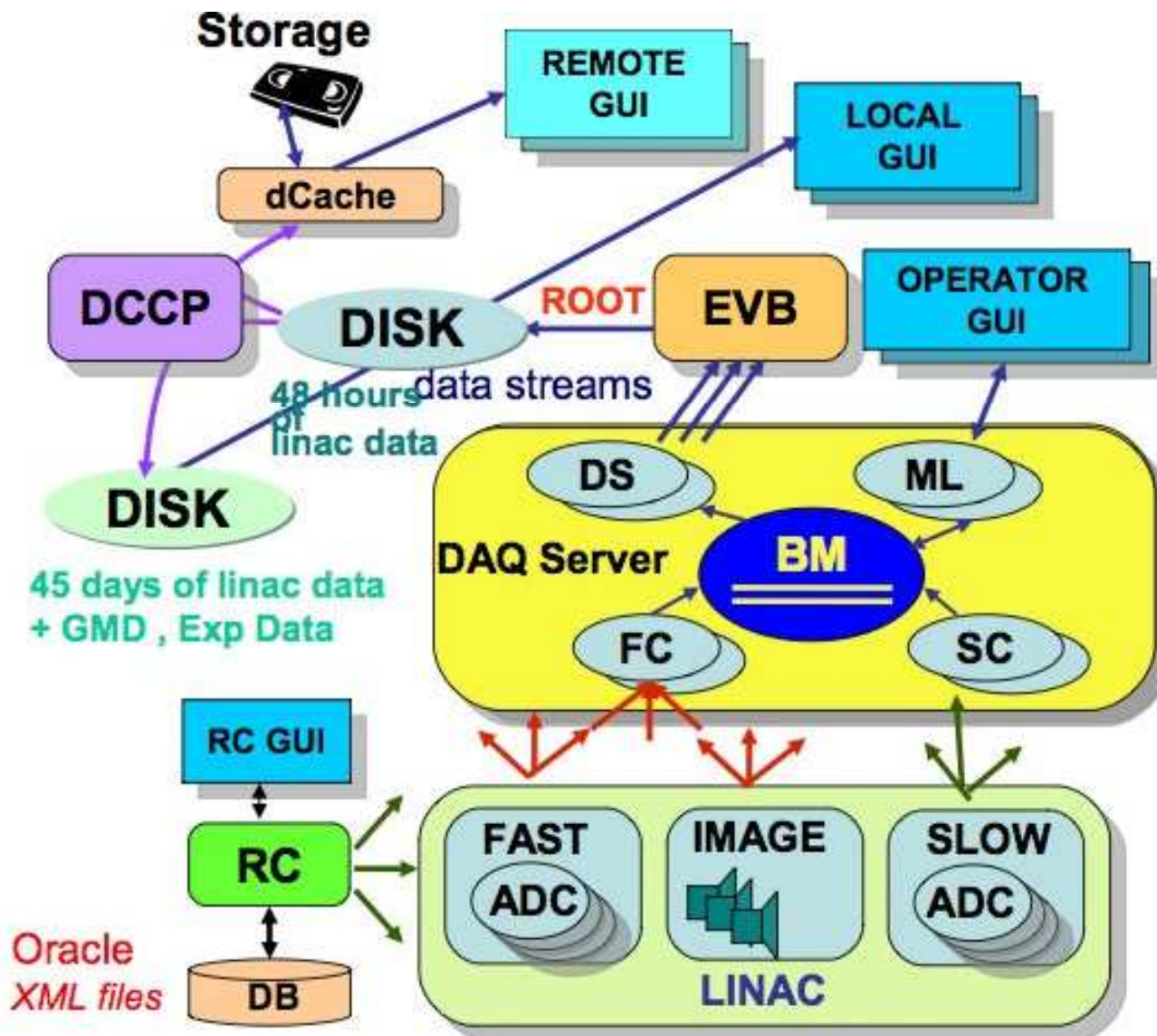
XML based Alarm and Info Server Architecture



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