



WG5: Diagnostics and Controls



Control System

Part 1: Hardware

Part 2: Software



WG5: Diagnostics and Controls



The XFEL Control System

Software

9. 5. 2006

Kay Rehlich



Motivation



The challenge:

- Dealing with > 100 MB/sec. from hundreds of distributed computers
- Design a system now with a software technology that is not obsolete 2012



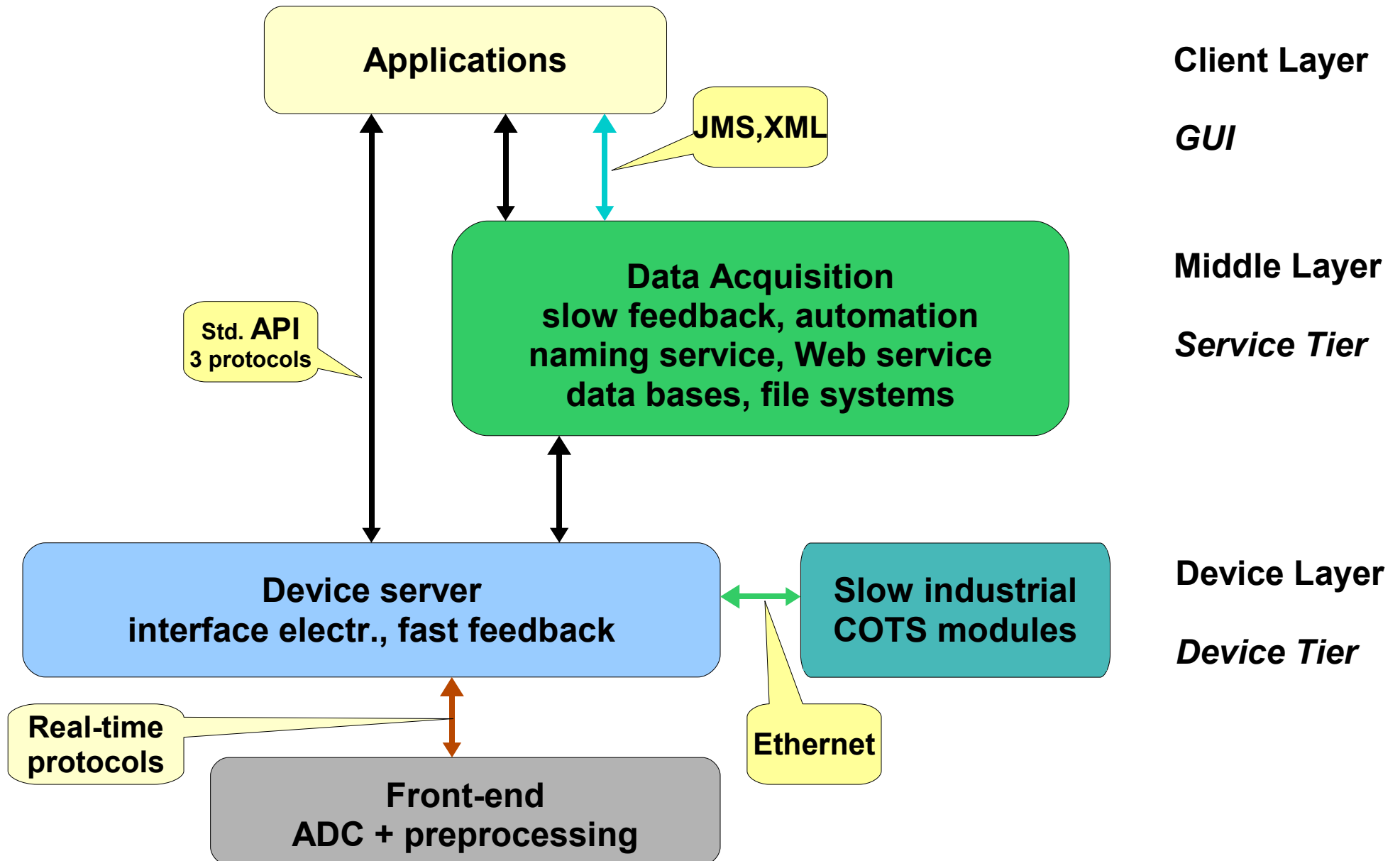
Content



- Overview
- Architecture
- Data Acquisition
- Applications
- Conclusions

Presentation of the actual state of the discussion

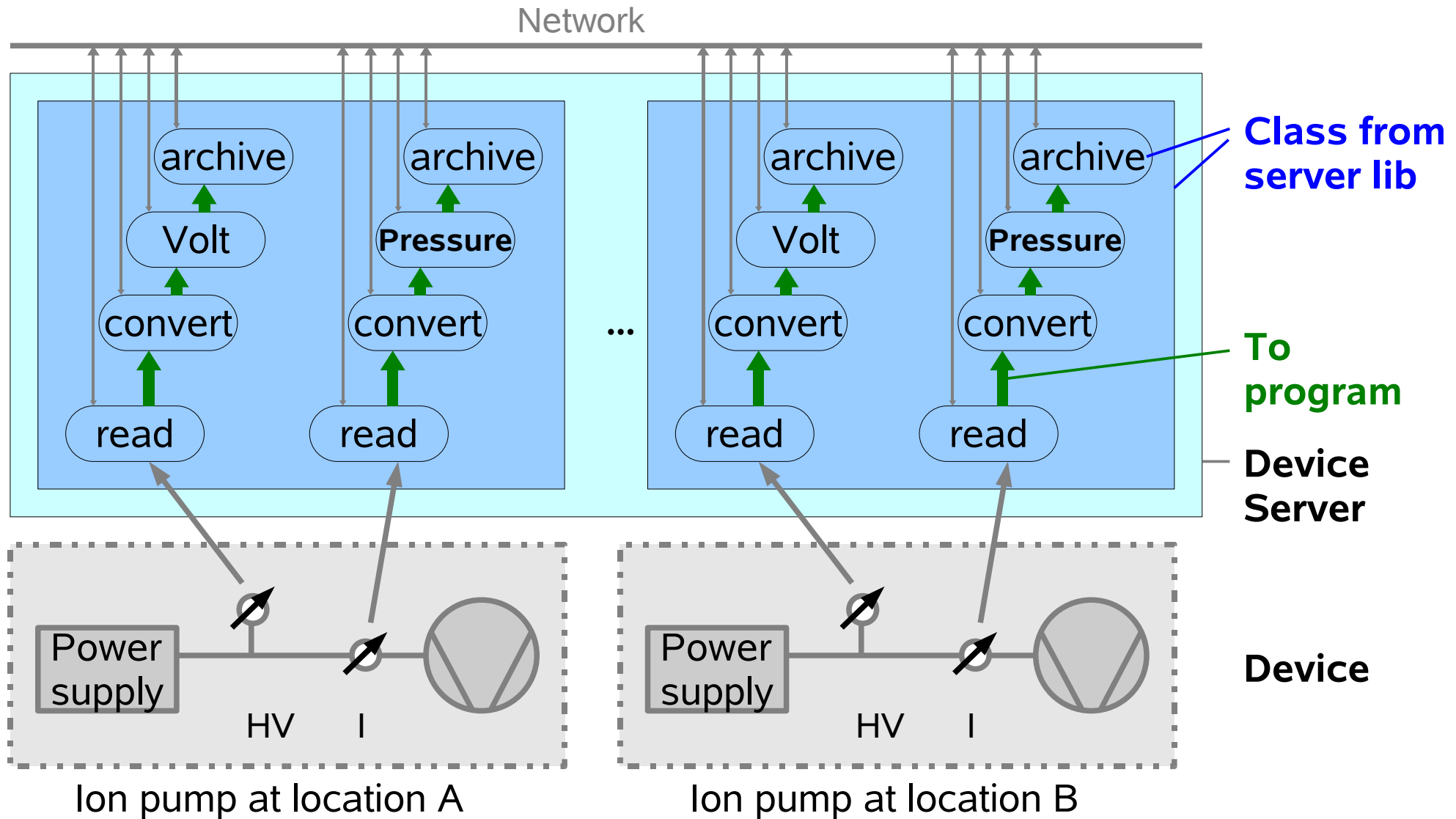
- **Will be based on the FLASH control system DOOCS (Distributed Object Oriented Control System), new features will be added**
- **It is planned to use FLASH as a test-bed and to smoothly upgrade FLASH (keep XFEL / FLASH software compatible/maintainable)**
- **Object oriented approach:
device abstraction on 'device tier'
subsystem abstraction on 'service tier'**



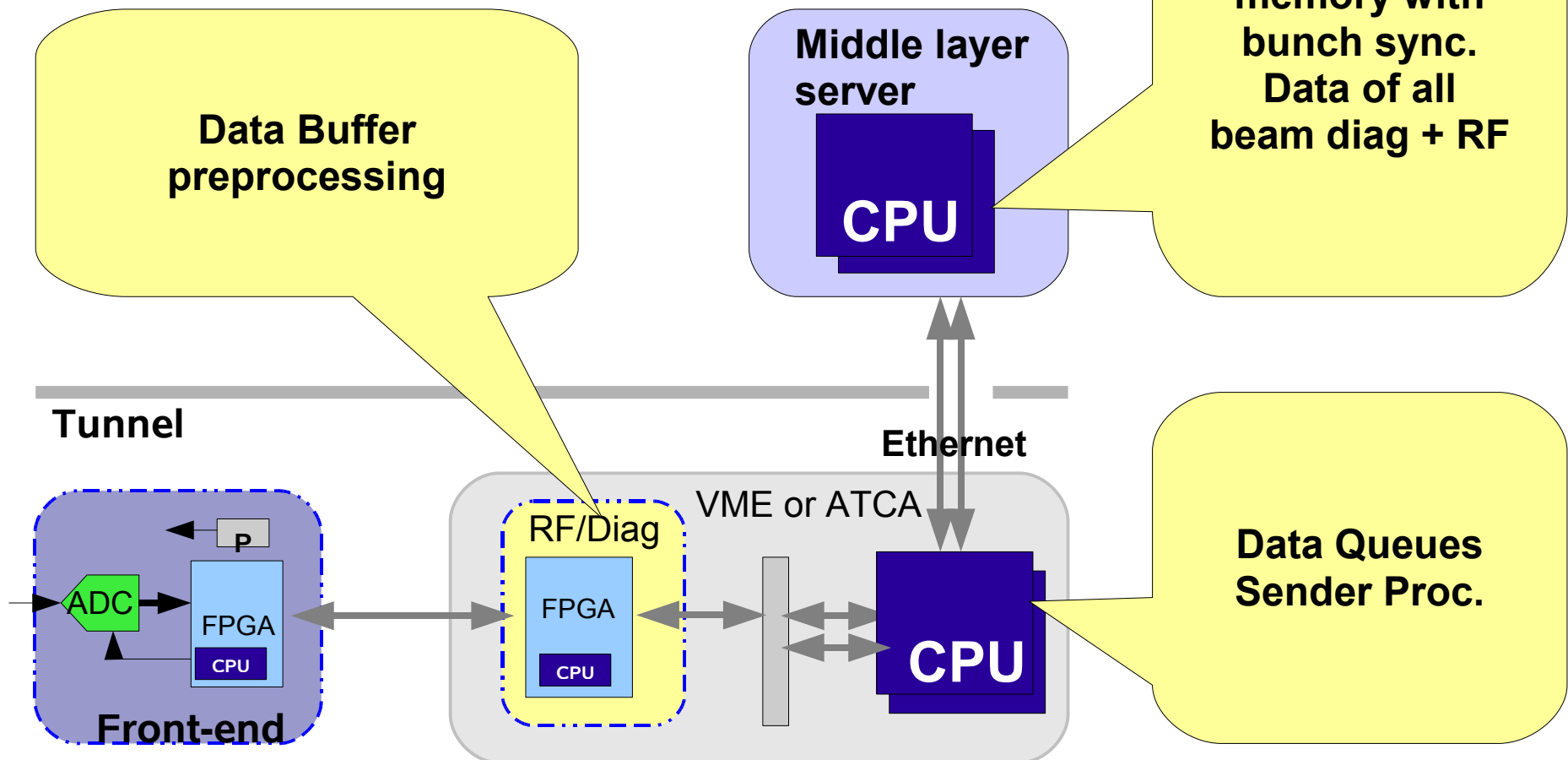
The **D**istributed **O**bject **O**riented **C**ontrol **S**ystem (DOOCS):

- Object oriented
 - Modern technology
 - Able to handle huge number of devices
 - Device objects on server and display
- Well defined separation of tasks in *three* layers
 - Device servers | middle layer servers | client applications
- Modular
 - Based on libraries (C++, in future JAVA for the clients)
- Self-contained device servers
 - Auto restart with previous settings
- All parameters on-line configurable
- Implements the required technologies for XFEL and FLASH

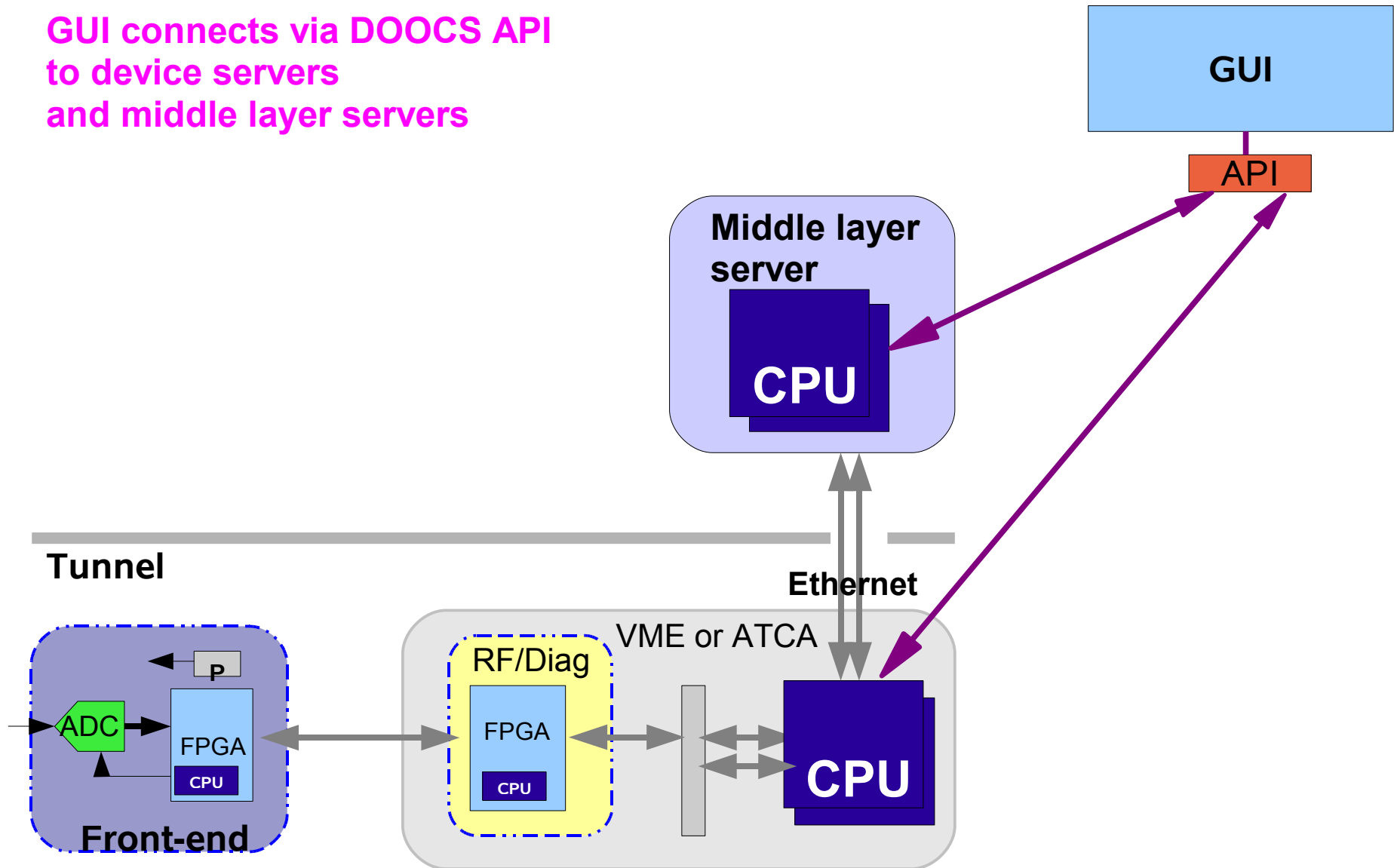
- Operating Systems: Solaris, LINUX, Windows (image server only)
- Hardware platforms: VME/ATCA CPUs, PCs, SPARC servers
- Used field busses are:
 - CAN, ProfiBus, RS232, GPIB, TCP, SNMP
- 150 different device servers exists @ FLASH
- Each server has a local configuration file that is updated
 - Reboot or server restart restores last state
- A watchdog process checks and [re]starts the servers
- Security: user ID and group ID for write protection
- Server library:
 - provides local archiving
 - Multi-threaded for 'soft' real-time and non blocking



Collects data from ALL bunches for:
slow feedbacks, correlations,
off-line analysis (e.g. trips), experiments ...



GUI connects via DOOCS API
to device servers
and middle layer servers

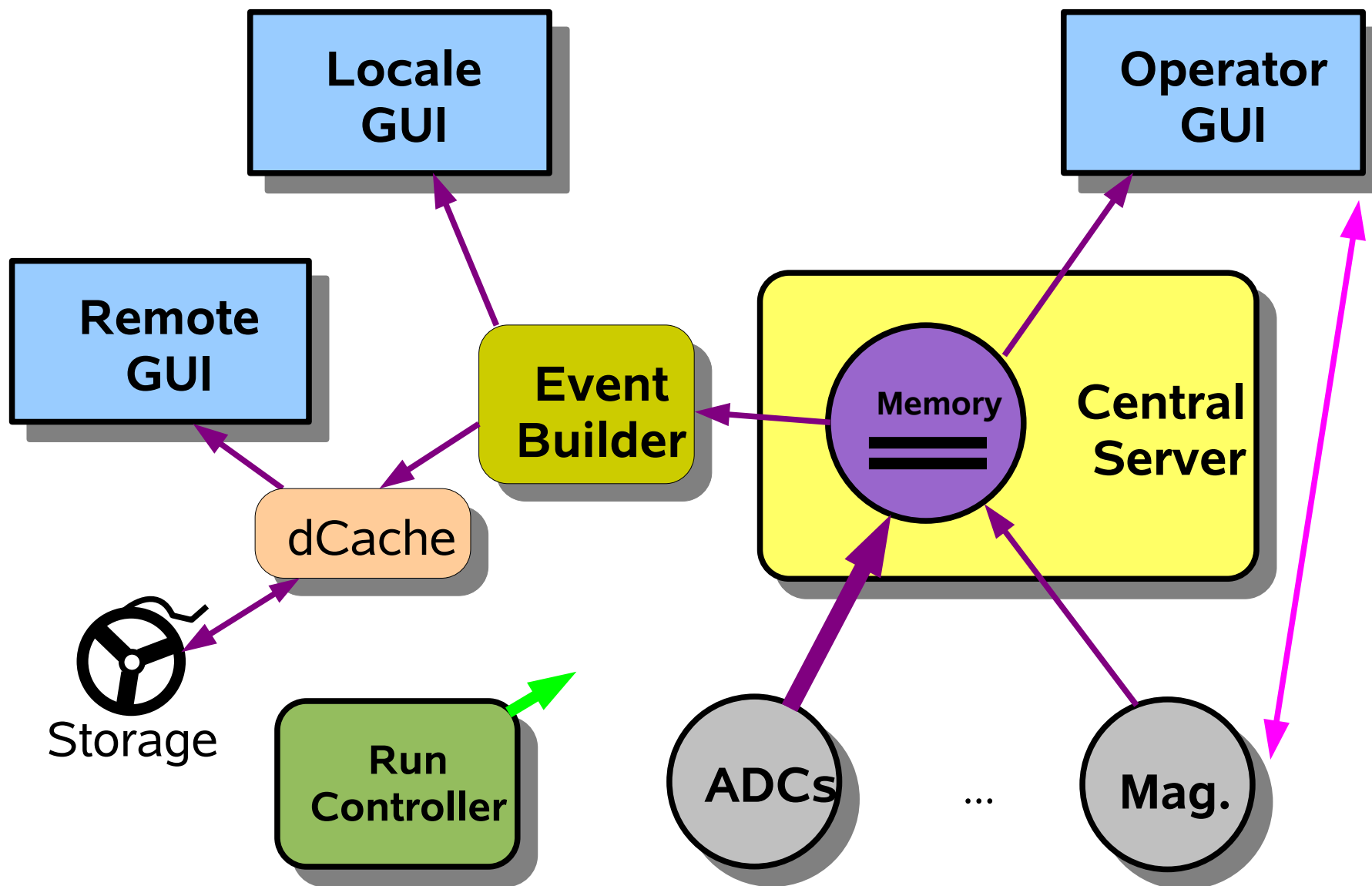


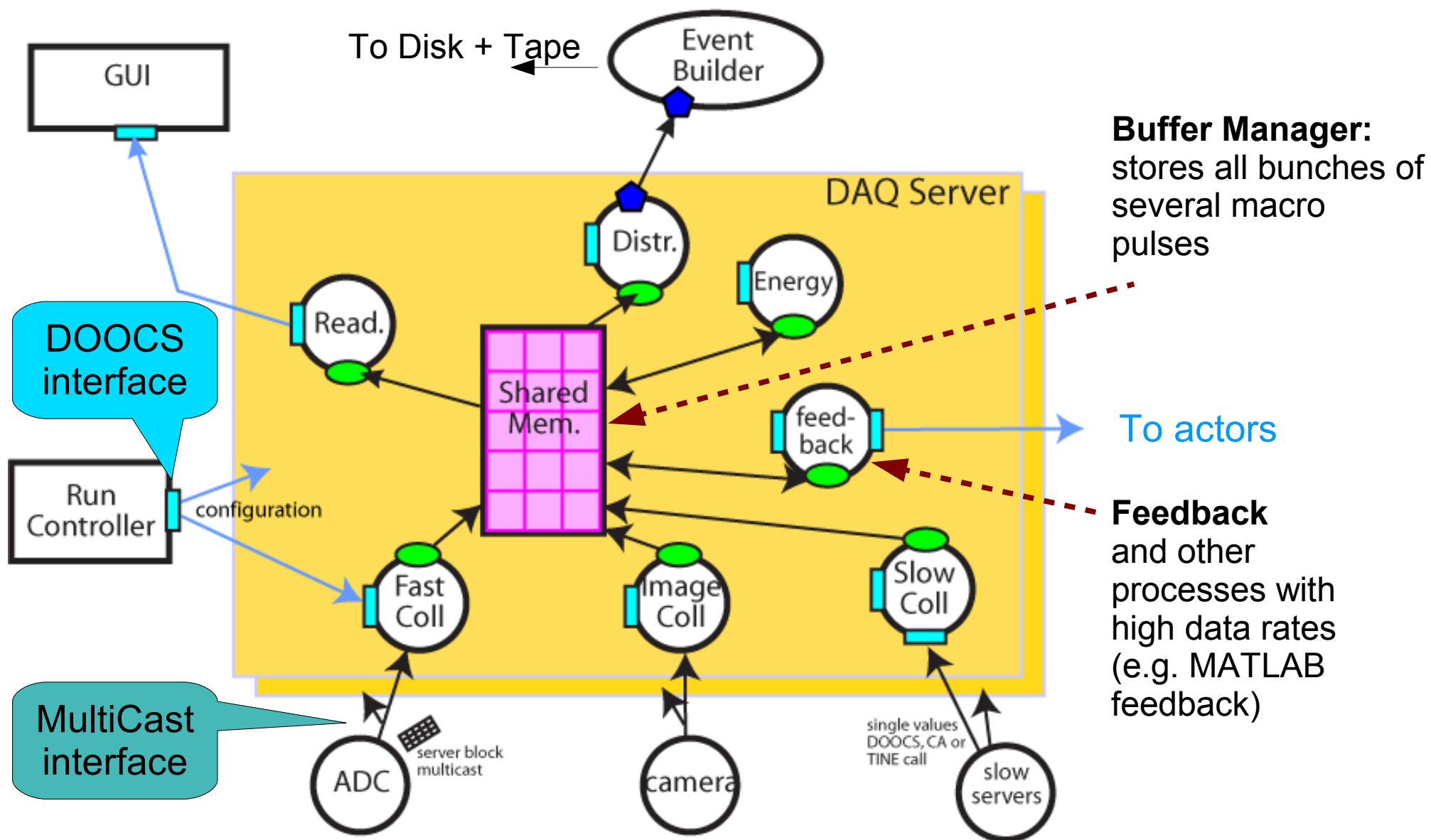
Linac data rates (rough estimate):

	XFEL	ILC	FLASH
BPM stations	500	2000	71
Toroids, BLM	300	1000	80
* bunches * rep rate	30000	16000	72000
LLRF cavities	1000	20000	40
* samples * rep rate * values	30000	10000	60000
		Probes only	
Tot max data rate [MB/s]	378	1831	79
DIAG [MB/s]	149	305	61
RF [MB/s]	229	1526	18

- **Tested rate @ FLASH: >30MB/s** (CPU: 12 Processors, SPARC)
- **Moore's law:**

2005	2007	2009	2011	2015
50	100	200	400	1600

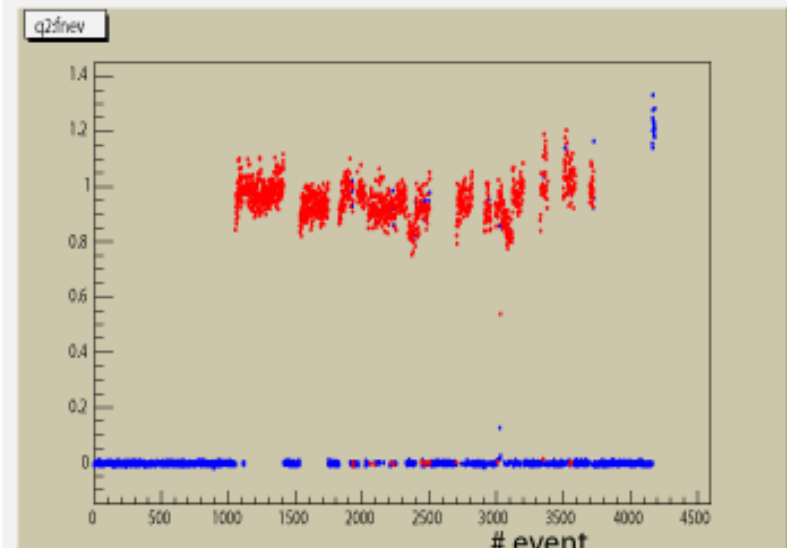
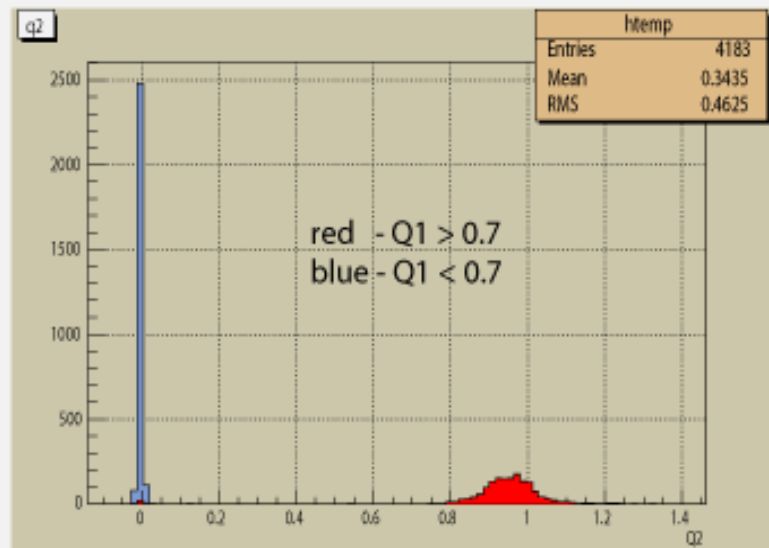
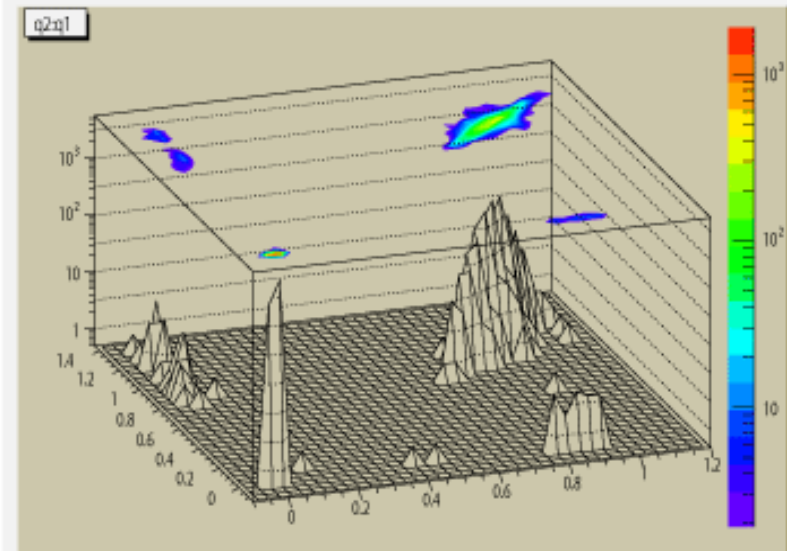
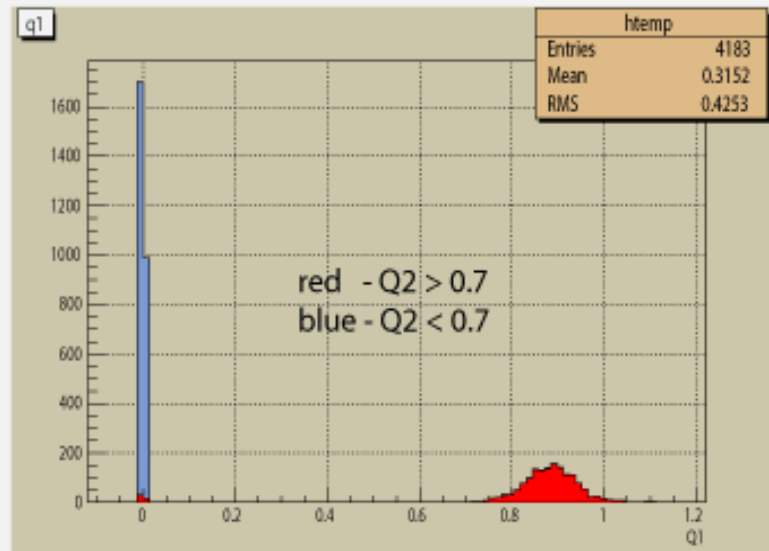




TOROIDS: Q1 - 1GUN, Q2 - 3GUN

CUTS: RED - $Q > 0.7$, BLUE - $Q < 0.7$

Plots
created
with
ROOT
tools

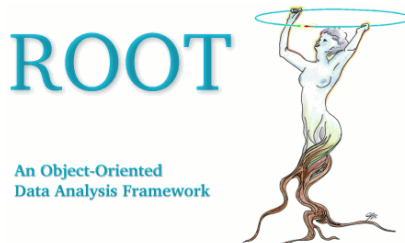




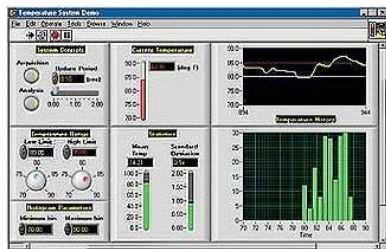
- ddd (DOOCS data display)
- setups and controls all devices
 - starts all other applications



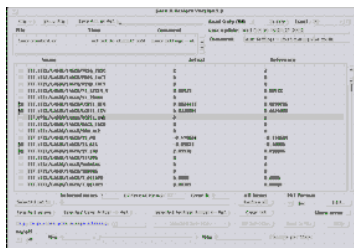
- MATLAB
- simulates e.g. the RF system
 - for writing ad-hoc applications



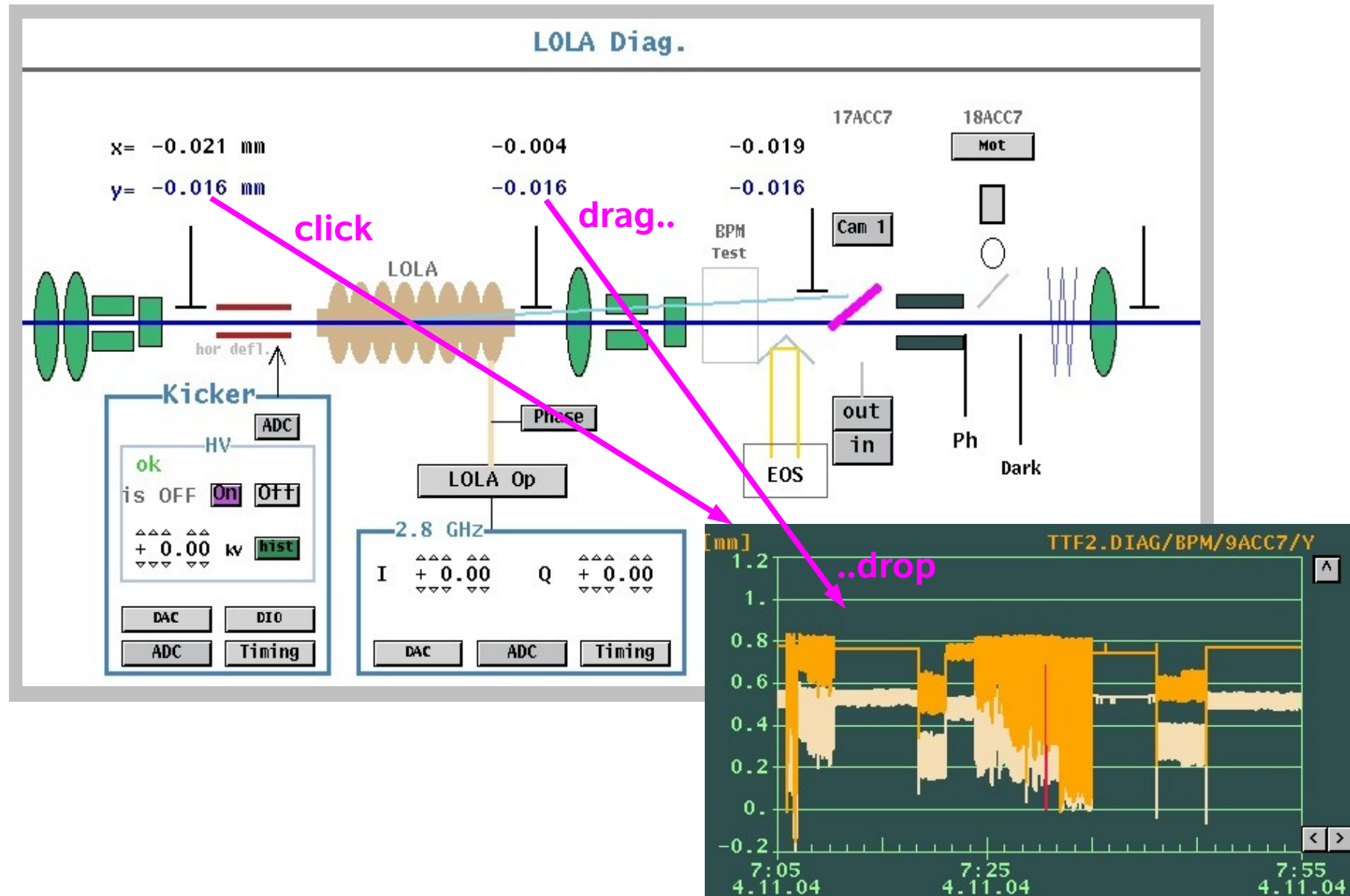
- ROOT
- displays the DAQ data
 - display and control: orbit, time-of-flight



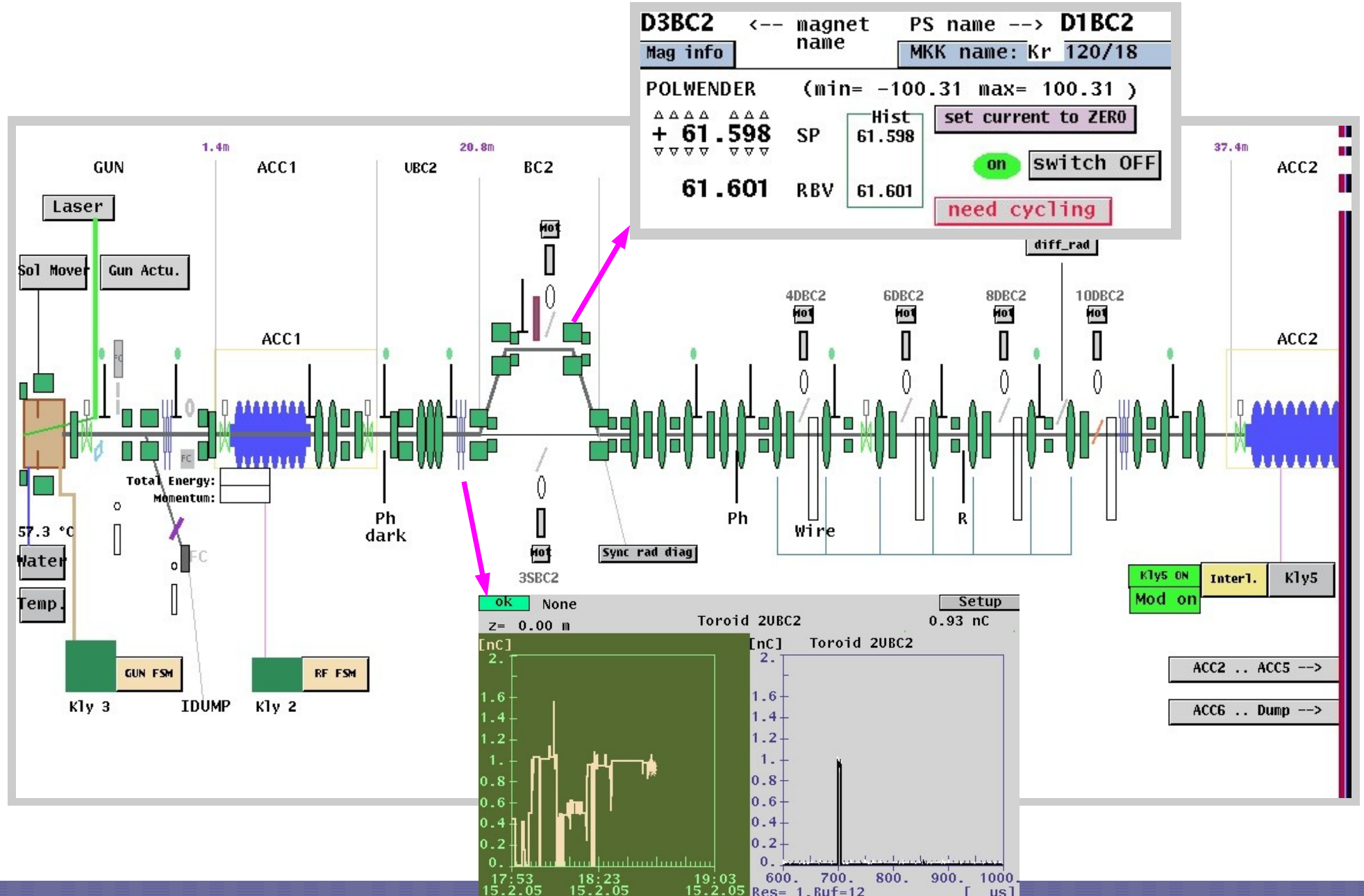
- LabView
- operates the OTR-monitors system
 - conditions cavities, operates test stands

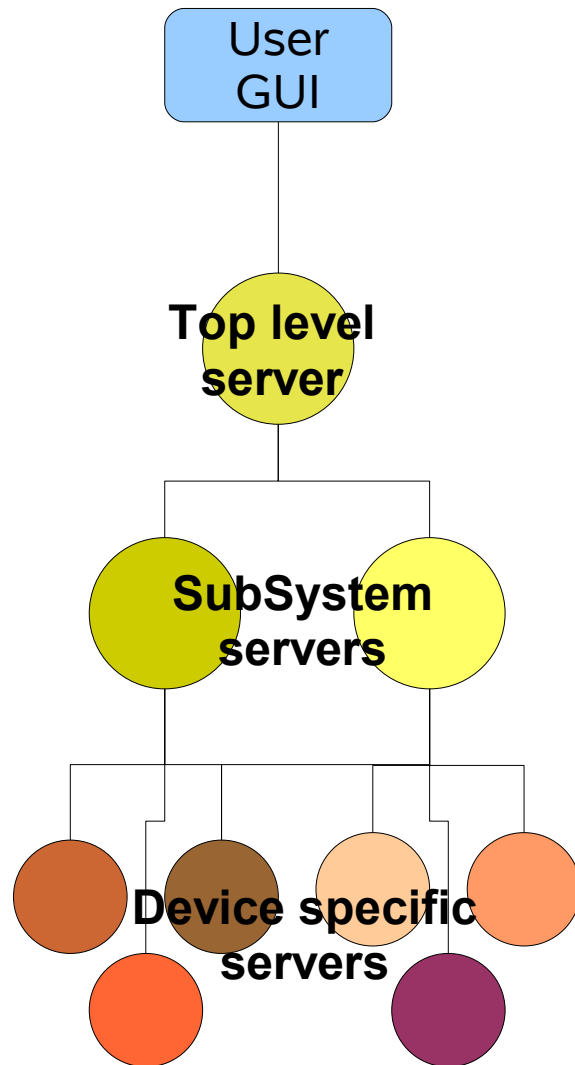


- 'Save & Restore' utilities
- saves and reloads linac settings
 - manages device configurations

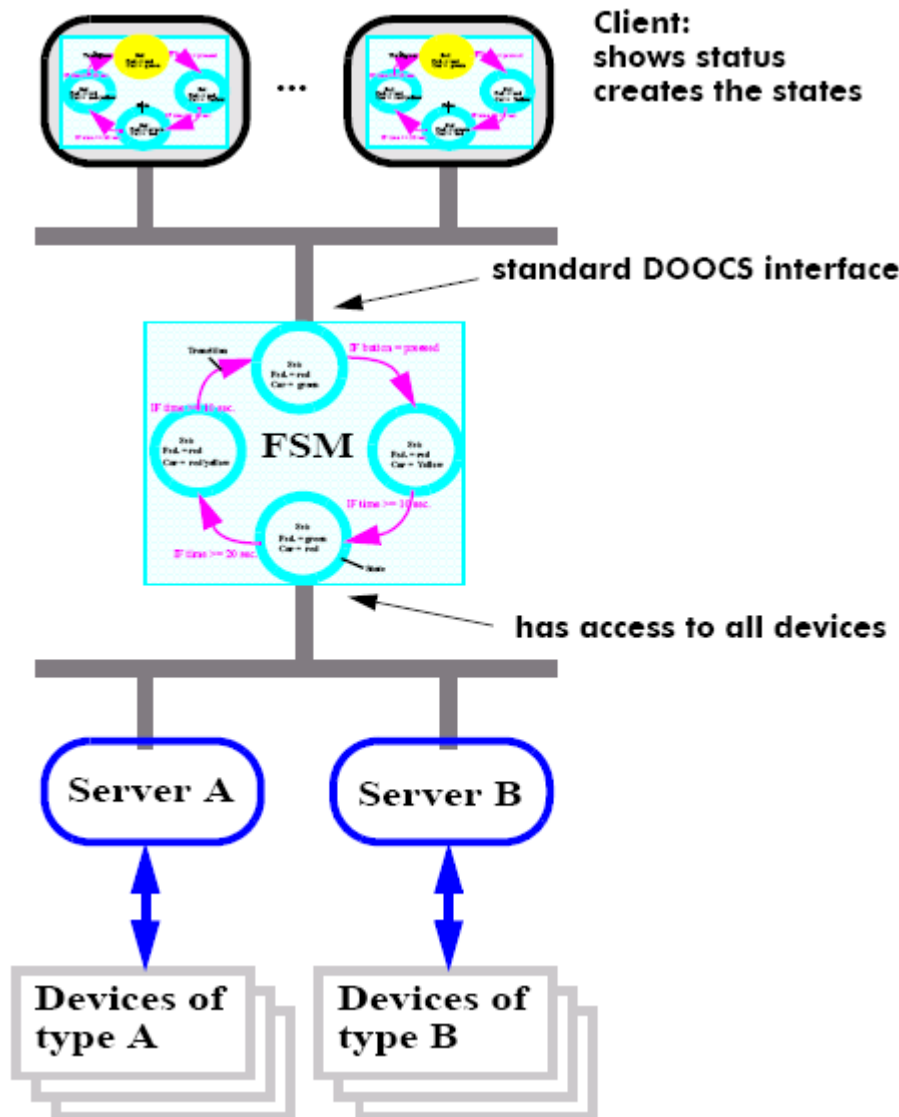


Injector Operation Panel





- Hierarchical structure
 - Grouping of functions
 - To reduce the complexity
 - To keep devices independent
- Finite State Machines and processes
 - Integration of high level tools (e.g. MATLAB procedures)
- Full control system integration



- Used for complex procedures e.g. LLRF
- Implemented on the middle layer
- Tool to create a FSM:
 - Graphical design
 - Code generator
- FSM is Multithreaded

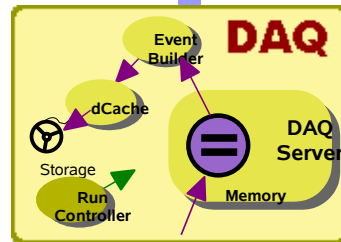
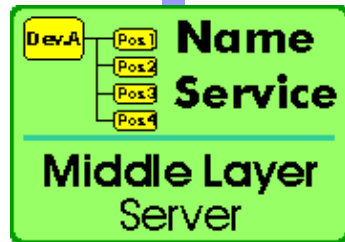
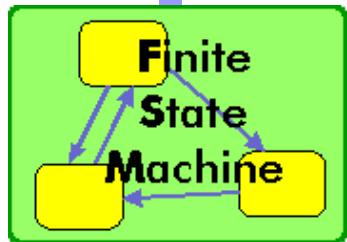
Overview



User
interface



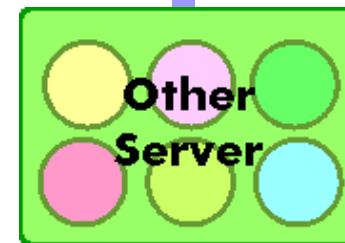
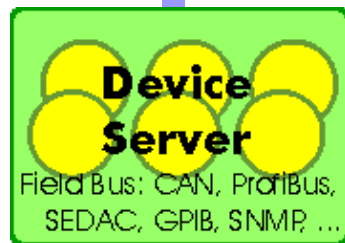
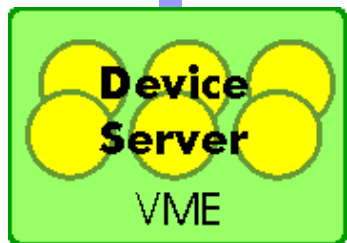
Program
interface



Middle
layer



Program
interface

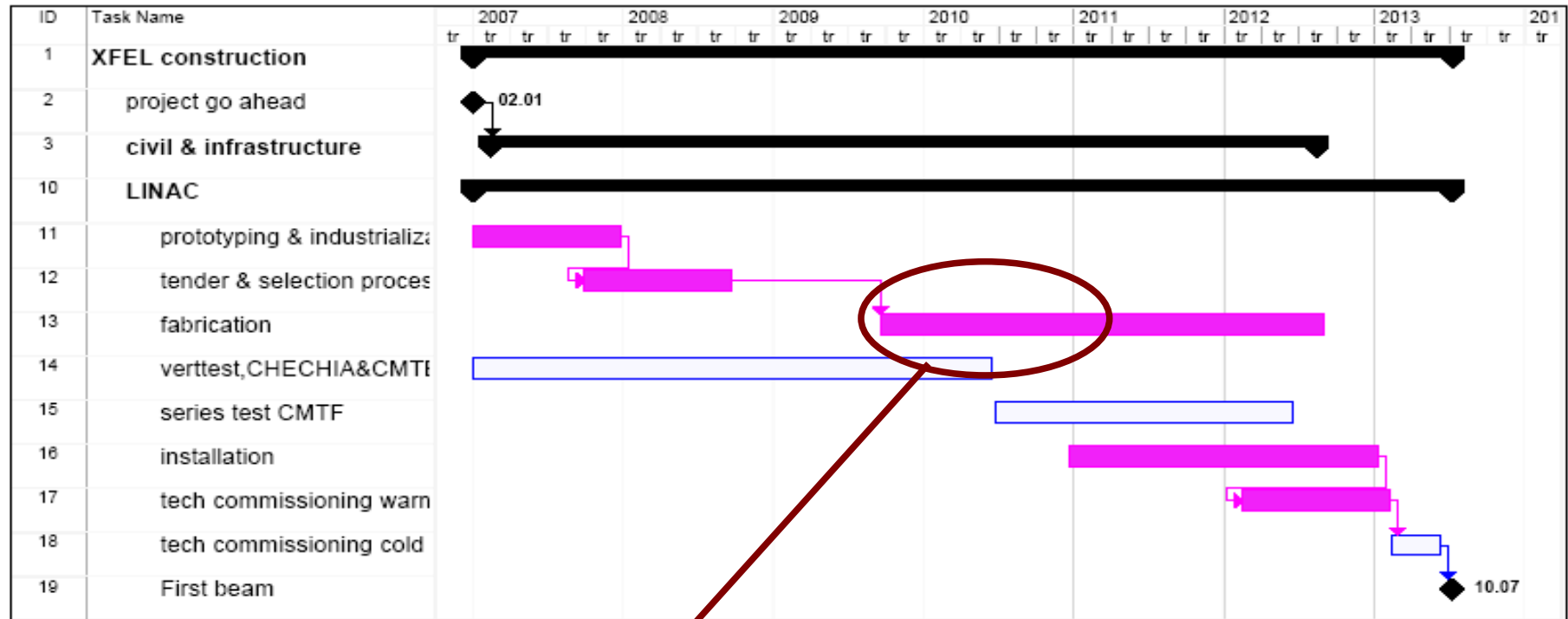


Hardware
interface

Hardware

- **DOOCS: runs @ FLASH, migration to XFEL**
- **Object oriented technology**
 - C++ and JAVA
- **DAQ: Novel combination and integration of an accelerator and HEP experiment control system**
 - Archiving of all bunches possible
- **Integration of commercial tools: e.g. MATLAB and LabVIEW**
- **Integration of tools from the HEP/accelerator community**
- **Must be available 24h, 7 days a week:**
 - 'self healing' by auto restarts
 - 'bump-less reboots'
 - Remote access by experts

More info: <http://tesla.desy.de/doocs>



- Main money will be spend on front-end hardware, server computers, and network
- 2006+: definition of
 - front-end standards etc.
 - Contributions from partners
 - Which items to buy or to design