ACCELERATOR CONTROL SYSTEMS

mark.plesko@cosylab.com

COSYLAB

Glossary 1 - Networking

COSYLAB

- LAN - local area network: directly connected computers
- Ethernet - physical (electric) communication
- TCP/IP - how to send/receive and assemble small data packets over network, in LAN or Internet
- switch - box which interconnects many computers
- fieldbus - simple low-cost connection of many sensors to one CPU

Glossary 2 - Electronics

- I/O - input/output: typical the place where digital data is transformed from/to something physical
  - example: printer, disk, ADC, etc.
- ADC - analog to digital conversion
  DAC - digital to analog conversion
  - standard ranges +/-10V, +/-5V or 0-10V, 0-5V
  - resolutions from 8 to 16 bits
  - precision usually worse than resolution !
- Binary or digital I/O - only two levels, for switches, etc.
  - beware of different levels: TTL (5V), 24V
  - open collector: the current for switching is not provided

Glossary 3 - Interfaces

COSYLAB

- interface - the agreed way how two different things fit together (hardware and software)
  - don’t forget about the connector!
- RS-232 - simple serial interface used in PC world
  GPIB - general purpose interface bus
  - many commercial devices (even oscilloscopes) use one of these
  - much work to interface, because software commands are not standardized - each device has own, often insufficiently documented
- API - application programmer’s interface: a set of library functions doing frequently needed actions

Glossary 4 - Hardware

COSYLAB

- CPU - central processing unit: heart of the computer, sometimes used for whole computer
- microcontroller - microprocessor with integrated I/O
- FPGA – Field Programmable Gate Array: millions of logic gates (AND, OR, etc.) that can be configured and connected with a „programming language“ - VHDL
- DSP - digital signal processor: specialized microcontroller doing fast Fourier and other transformations
- front-end: all that stands in front of the hardware
  - VME - standard format for I/O modules and processors
  - PLC - programmable logic controller: diskless microprocessor following simple logic instructions
- workstation - high performance desktop computer

Glossary 5 - User Level

COSYLAB

- GUI - graphical user interface (buttons, menus)
- SCADA - supervisory control and data acquisition: commercial CS, with GUI building tools and software bus
- database - where data are stored and retrieved efficiently
  - RDBMS - relational database management system: a powerful database with data arranged in separate tables, linked by defining relations between columns
  - SQL - structured query language: the most common standardized language used to access databases.
- application - one or more programs for a specific problem
  - usually includes GUI for user interaction

Glossary 6 - Programming

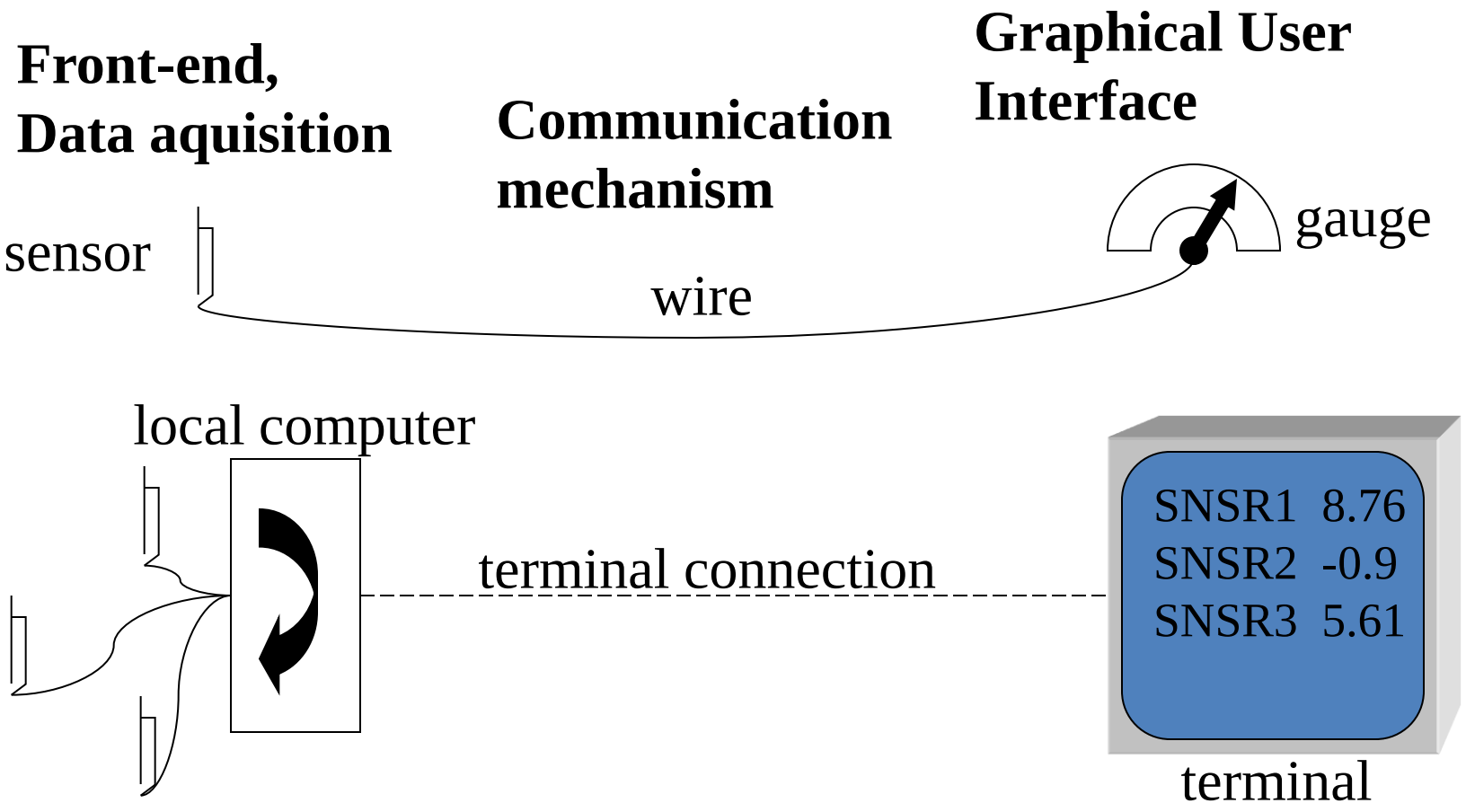
- synchronous - make request and wait for response
- asynchronous - response comes later, independently
  - polling - periodic checking if response is ready
  - callback - a response sent asynchronously, interrupting the receiver
- scalability - design how a CS is applied to large systems
- real-time - a process **must** finish in a given time
- OO - object oriented: programmer defines objects (structures) which exchange messages (function calls)
  - very similar to “normal” way of thinking
  - efficient approach for design, “software reuse” and upgrading

Let‘s first define what is a CS

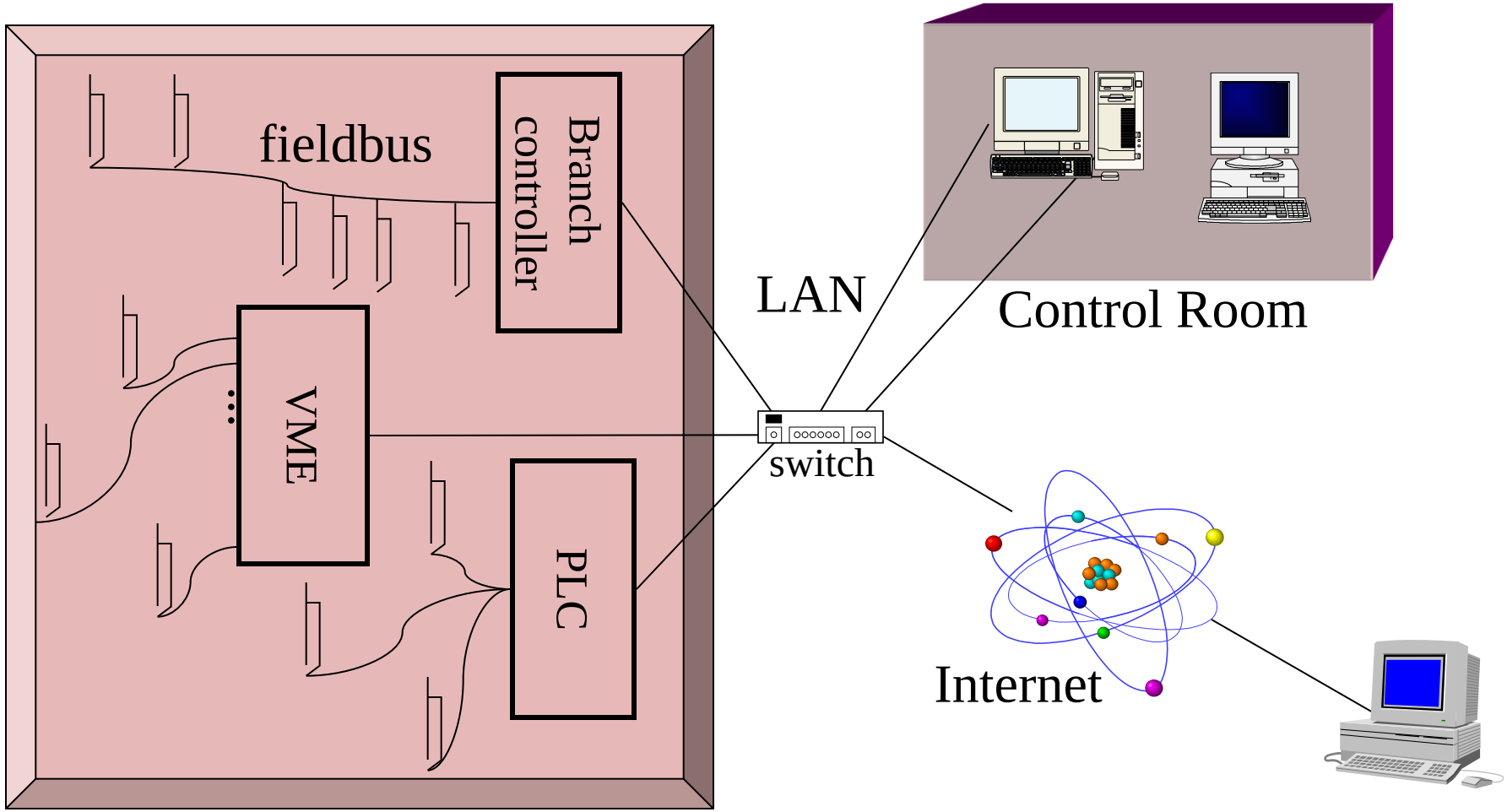
COSYLAB

- A tool that allows to operate the complexity of the accelerator from a single location
  - ~~Convert physical properties to electric (analog) signals~~
  - convert analog signals to digital values
  - transfer digital values over network
    - Access remotely each single device
  - Execute complex processes (injection, ramping, tuning, feedback systems, safety, etc.)
    - Log and archive data, events, actions
  - Allow physicists to „run“ the accelerator with Matlab ☺

From the Analog to the Digital... COSYLAB

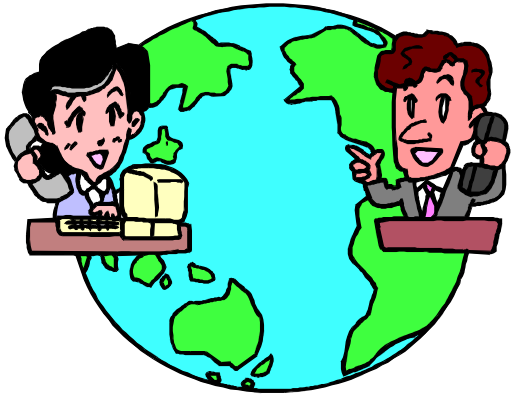


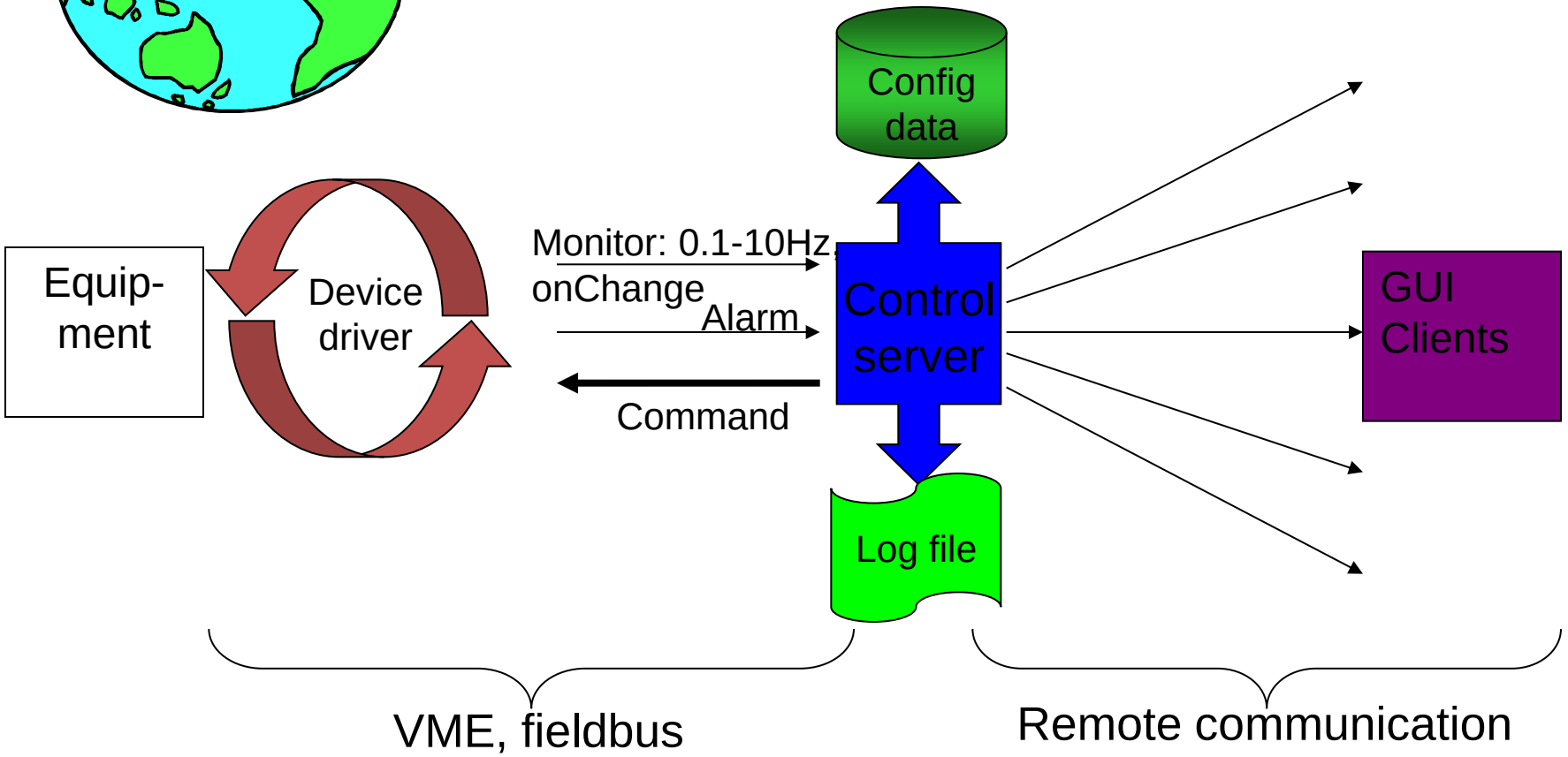
… to the Distributed Era



Data Flow

COSYLAB





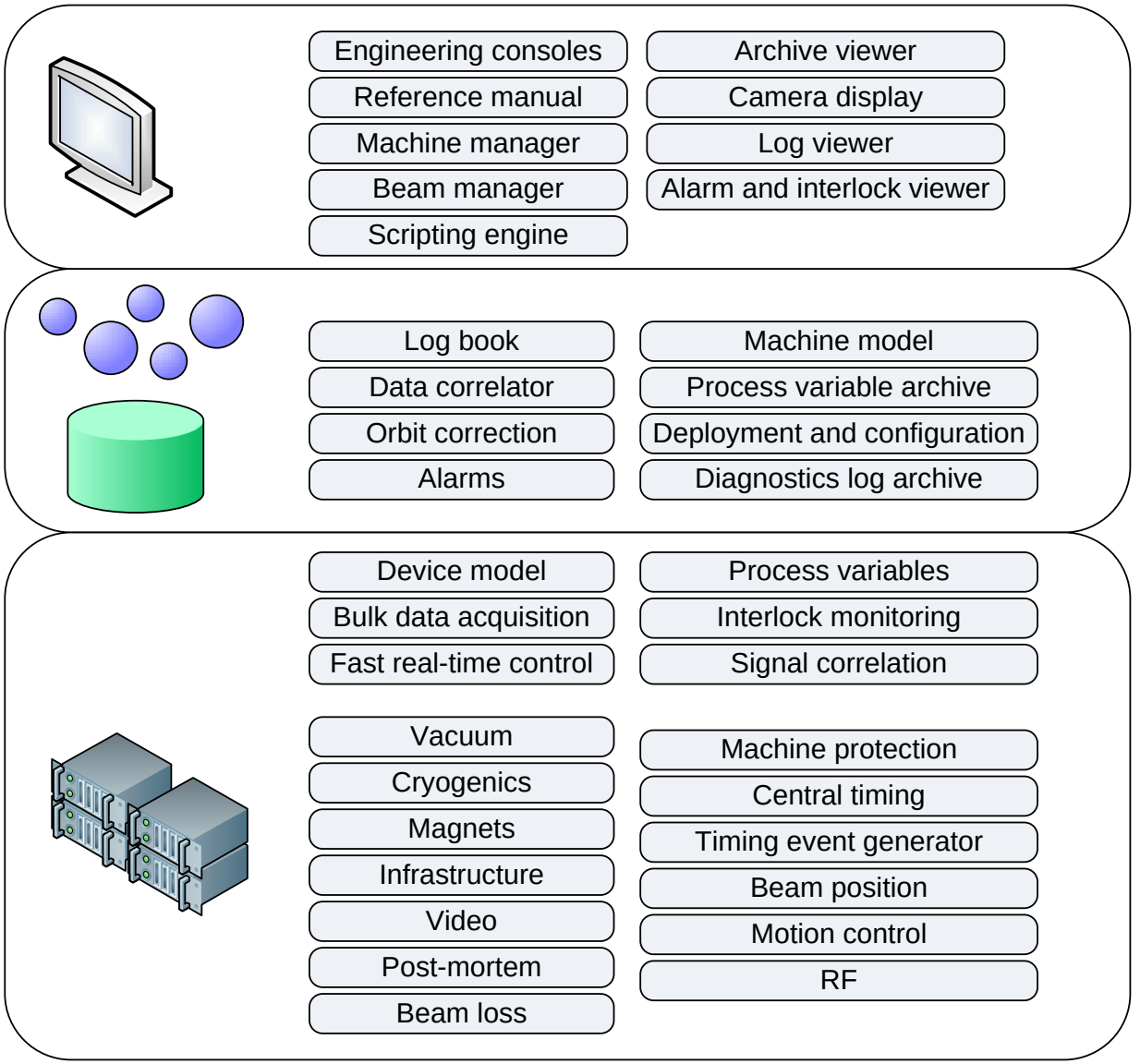
COMPONENTS OF THE CONTROL SYSTEM

COSYLAB

Main CS Components

COSYLAB

- What has to be in real time?
- Interconnection of components and services
- Which components are getting more important?

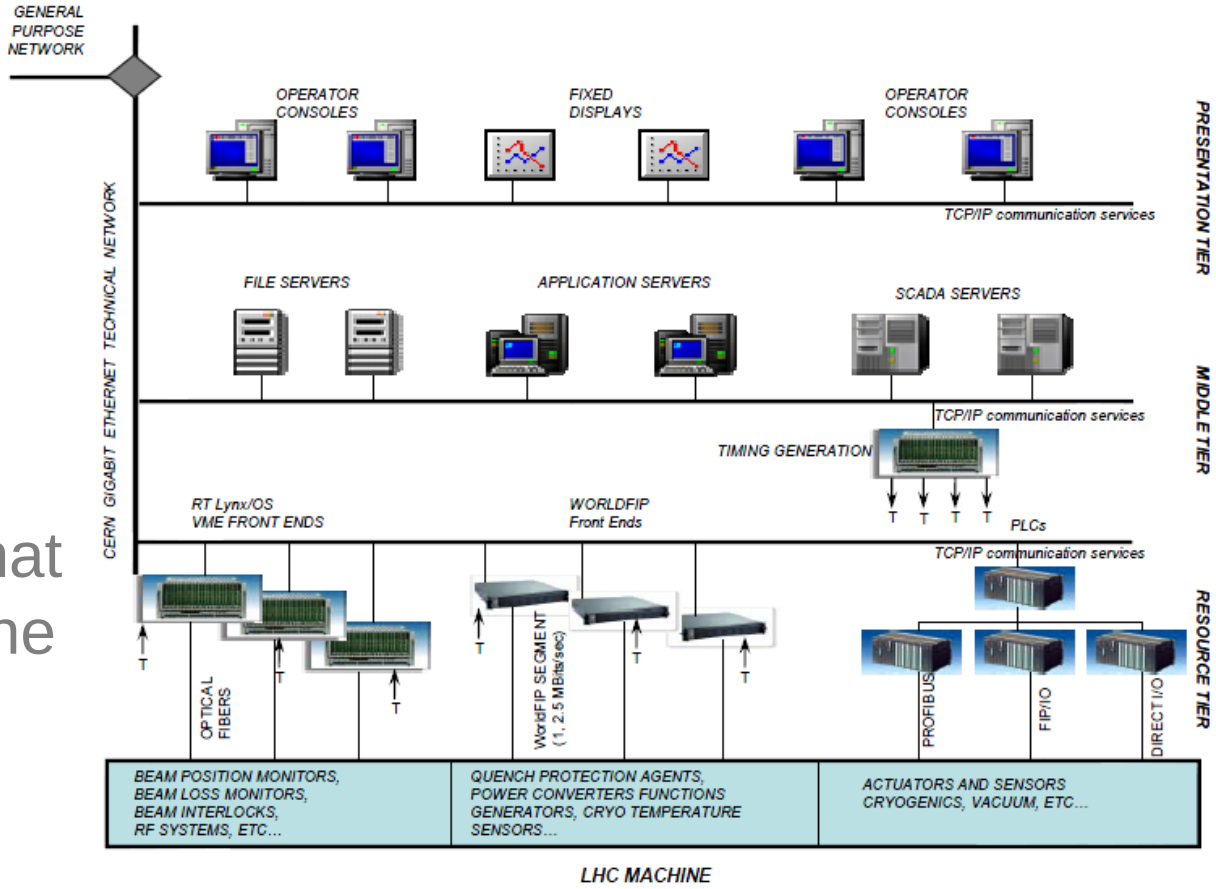


Middleware or software bus

COSYLAB

- What is their purpose, are they really different
  - **Yes** in terms of technical implementation
  - **Probably yes** in terms of performance
  - **No** in terms of what they provide for the CS

EPICS, CMW/FESA, TANGO, TINE, DOOCS, MADOCA,…

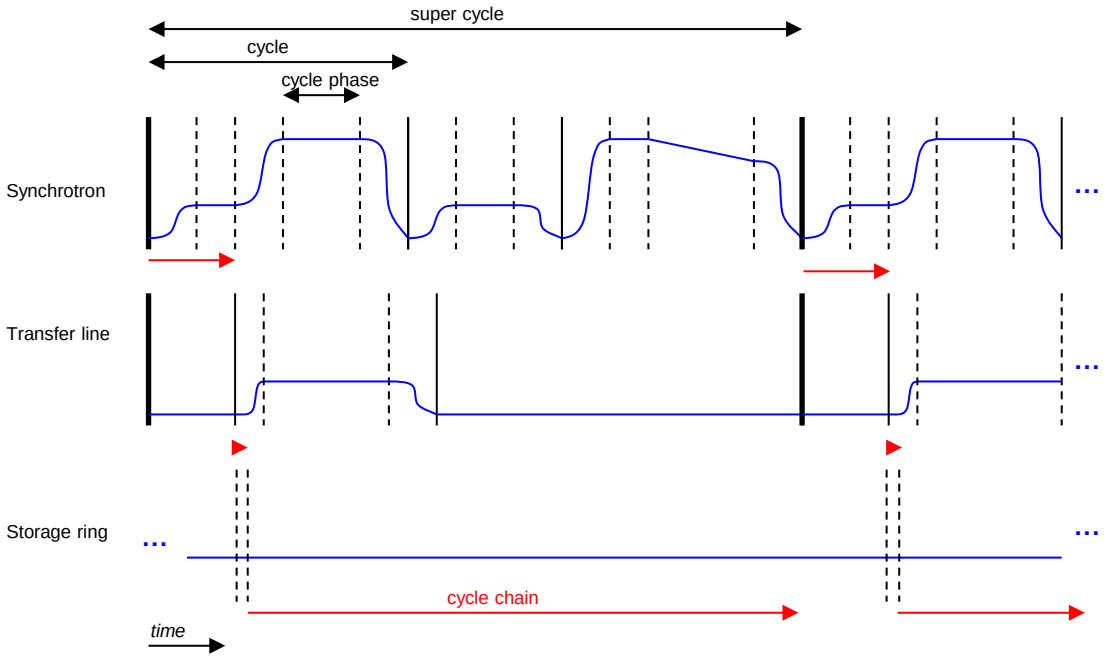


Why is EPICS so popular?

COSYLAB

- A very strong user and developer community
- A large number of supported devices, and a relatively small set of interfaces
  - e.g. Universal motion control record
- Lightweight on dependencies
  - Does not depend on relatively complex middleware
  - few central services that would be single-points-of-failure.

Timing System



Graphical illustration of the concepts cycle phase, cycle, super cycle and cycle chain.

- Takes care of real-time (BTW, what is this?)
- Must be a common solution for the entire facility
- Should be one of the first systems to be completed
- Provides
  - Triggers
  - Delay generators
  - Event distribution
  - Time distribution
- Accuracy to within 1 us, 1ns and for FELs even to 10 fs !

What does timing system include?

- it's like an orchestra;
  - a conductor (=timing master) instructs
  - performers (=timing receivers) to play out formerly agreed work

…the better in sync the better it sounds

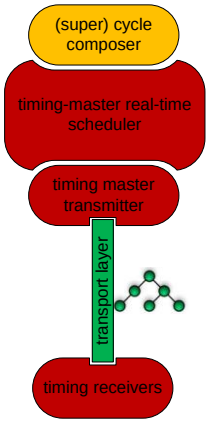
layers from composer to performer:

- (super) cycle composer
  - a high-level app for building table/s and play conditions—defined by machine physics and beam requirements—and transforming into suitable format for **(SW)**

- timing-master real-time scheduler
  - for priority scheduling of the events to be pushed through **(HW-RT)**
- timing master transmitter **(HW-RT)**

- transport layer
  - for real-time distribution of events (also clock and possibly time) **(HW-RT)**

- timing receivers
  - for real-time reaction to received events; providing host with triggers/IRQs, clock, and time **(HW-RT)**



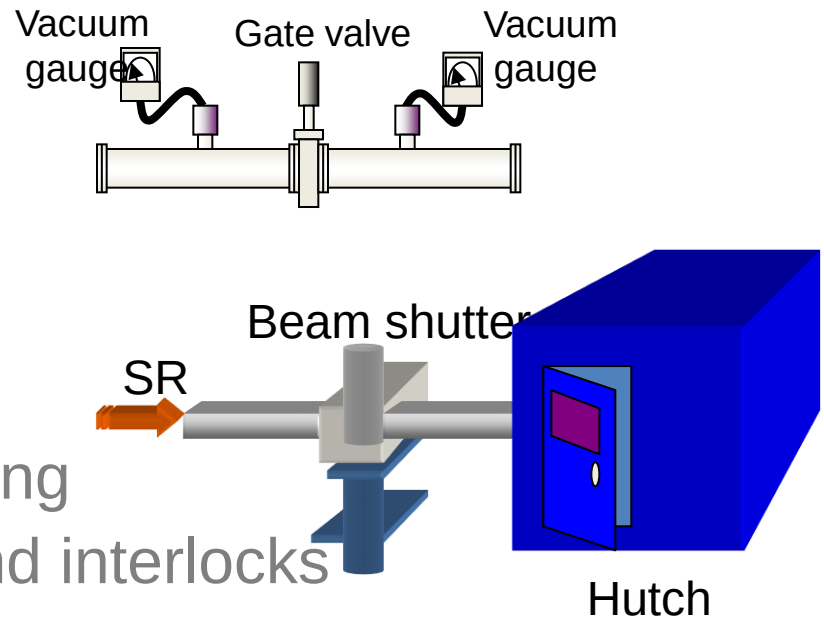
Still Plenty of Work …

COSYLAB

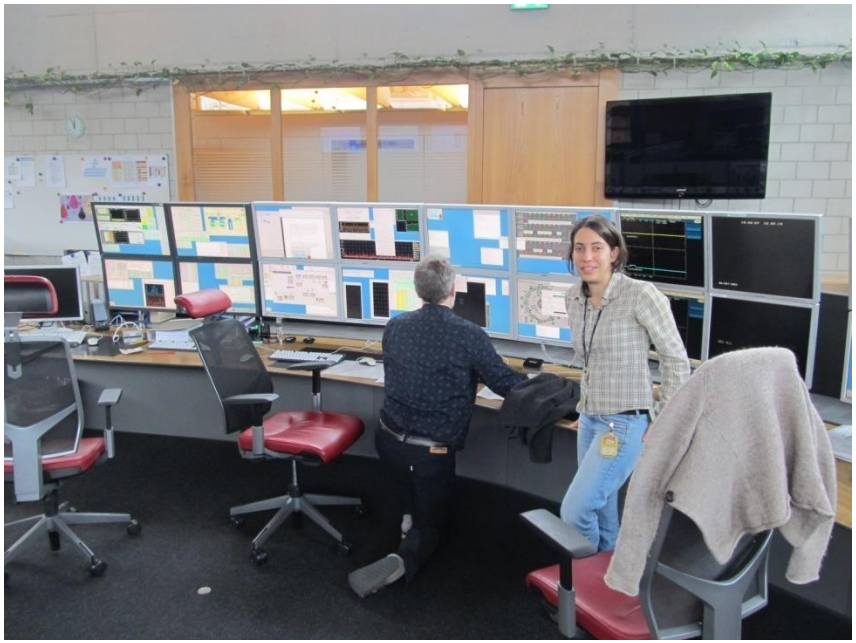
- interface I/O boards to the devices
  - where to draw the line:ADC/DAC, serial/GPIB
- write low level device processing
  - drivers, filtering, state machine
- configure middle-ware database
  - name, type, properties (limits, conversion, updates, precision,…)
- make graphical user interface (GUI)

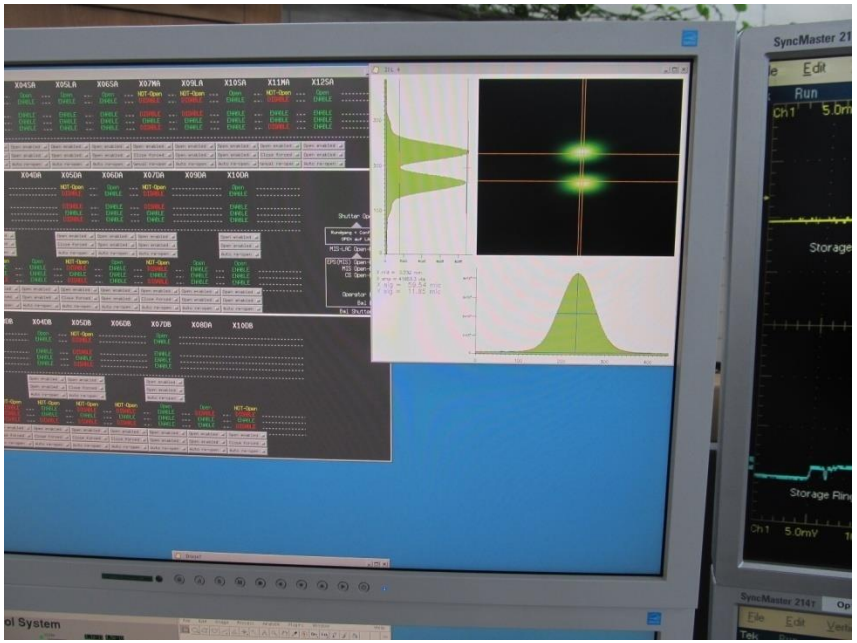
… Also For Others.

- Feedback systems - separate communication, dedicated hardware
- Safety systems
  - equipment safety
    - interlocks
  - personal safety
    - radiation monitoring
    - access control and interlocks



Control Room For Display…

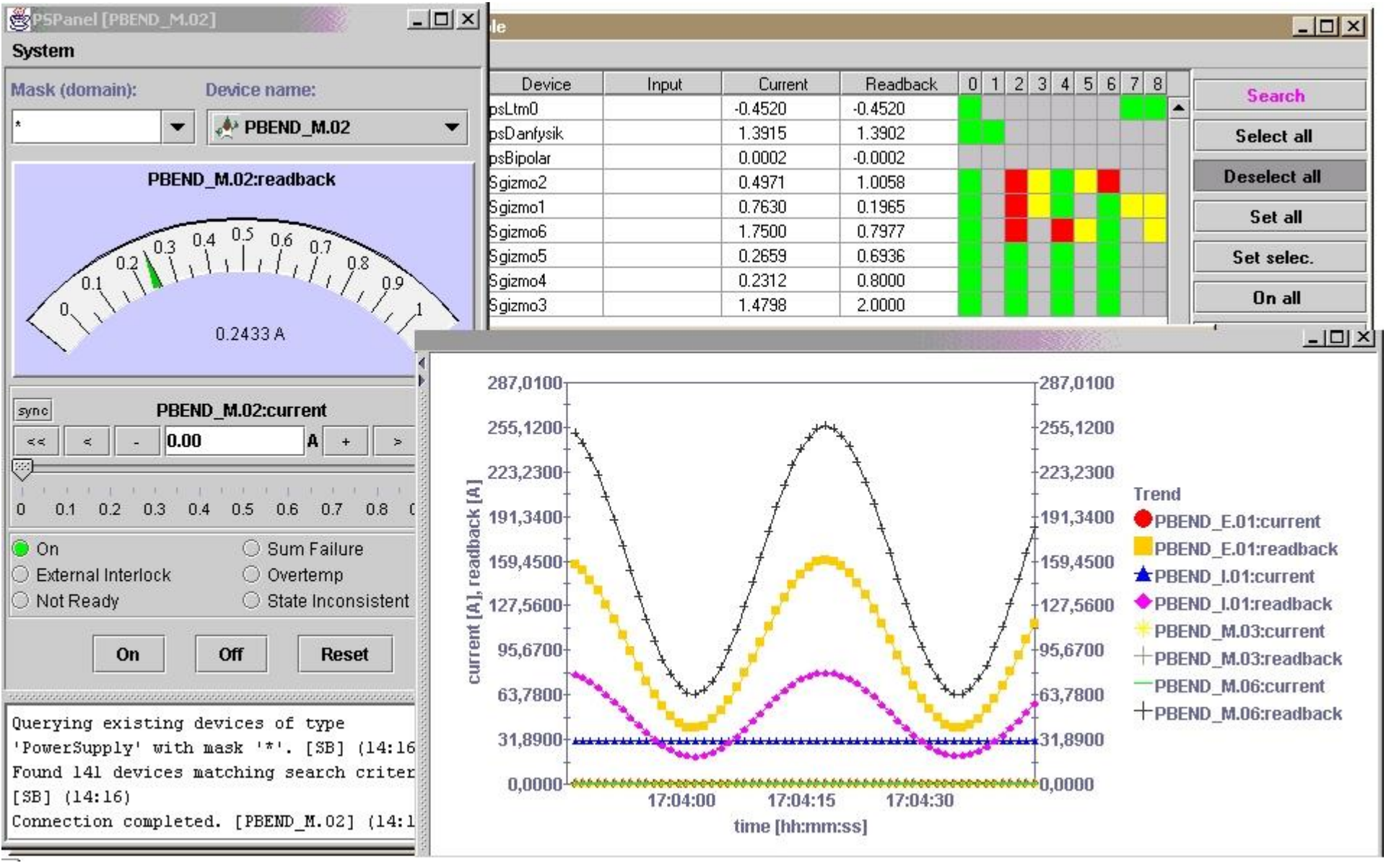




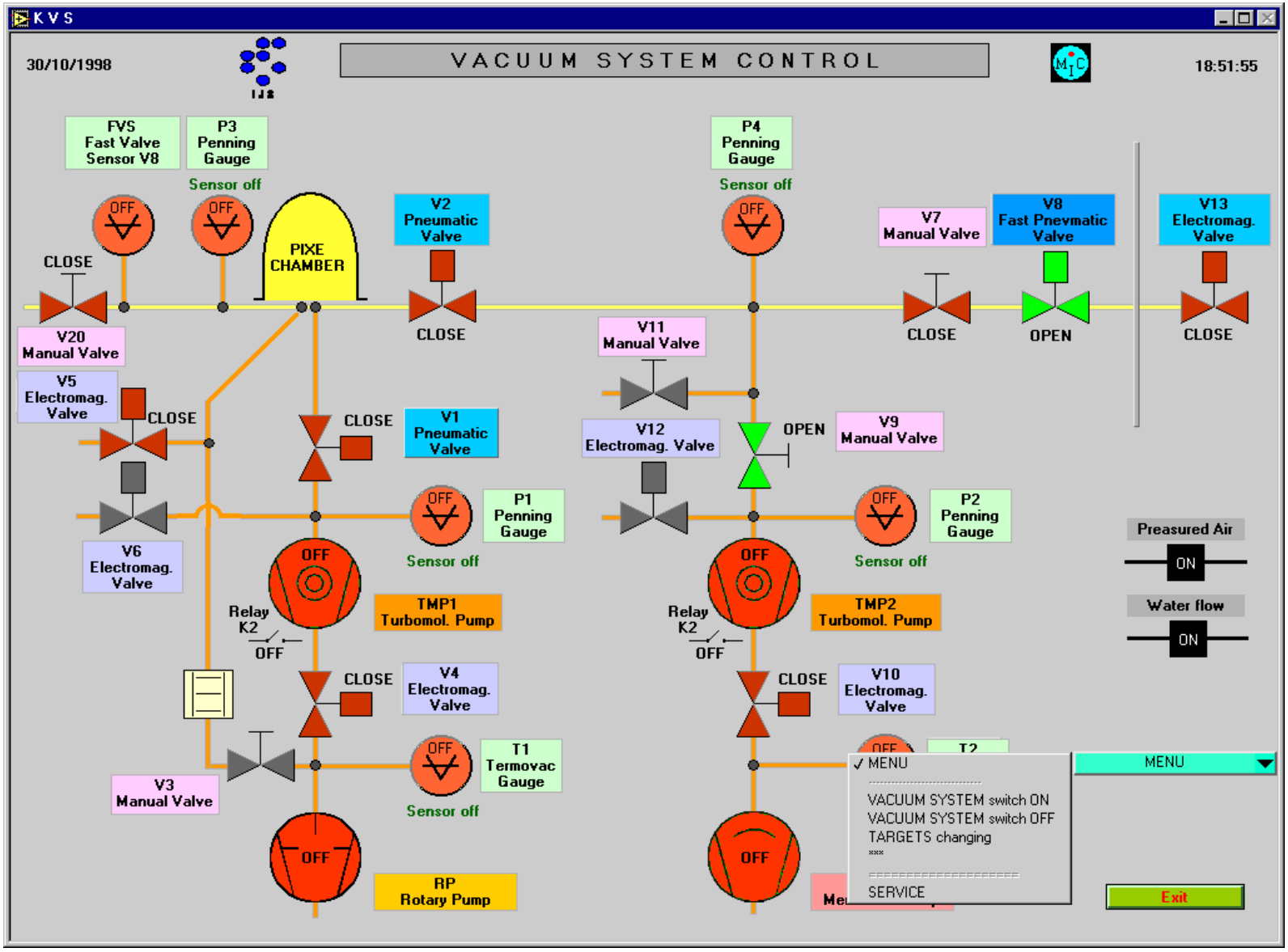
Success! Happy users. (above: first beam size measurements on new beamline)

…of Functional Panels…

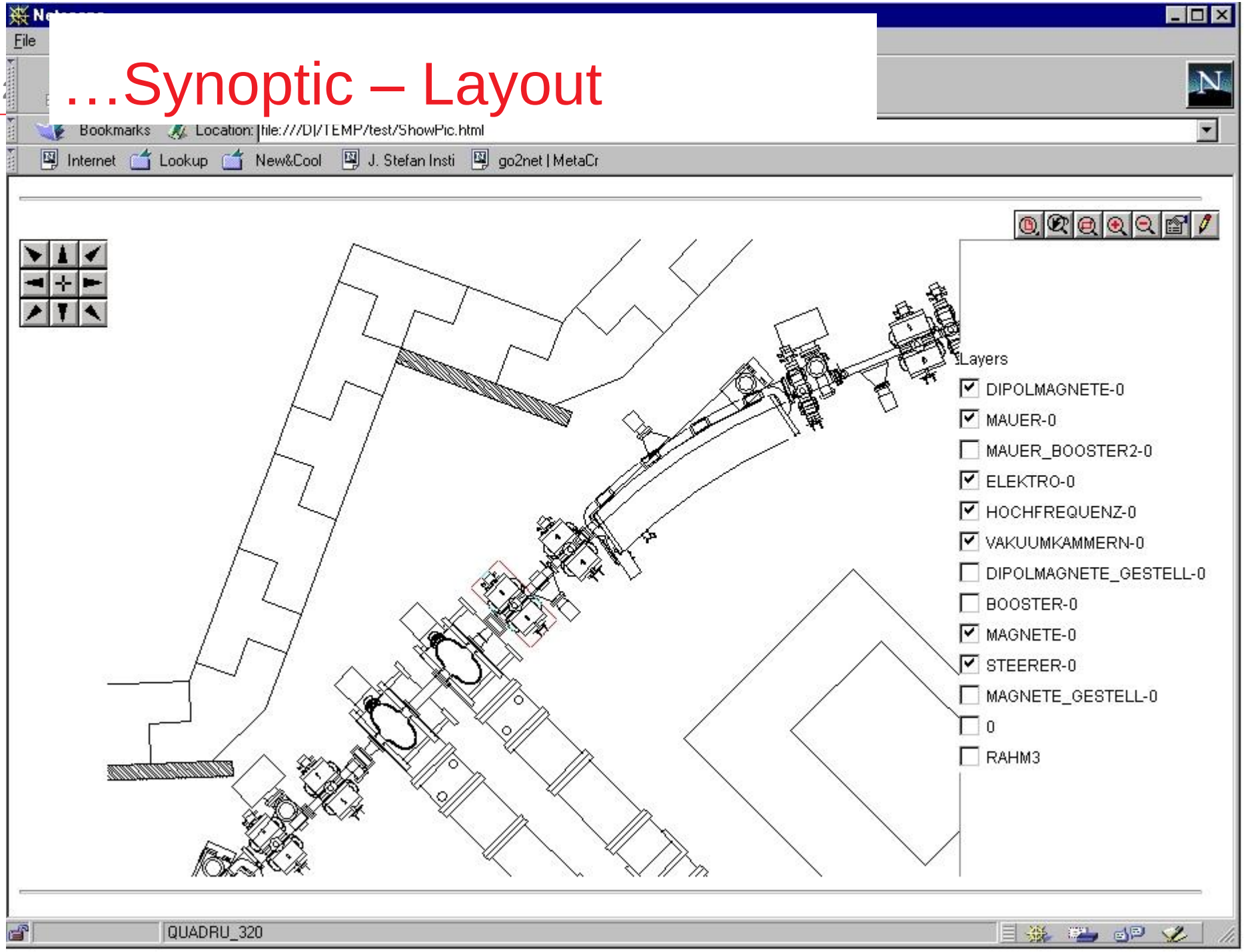
COSYLAB



…Synoptic - Process Diagram…



…Synoptic – Layout



Archiver

Alarms

Machine Protection and Interlocks 1/2

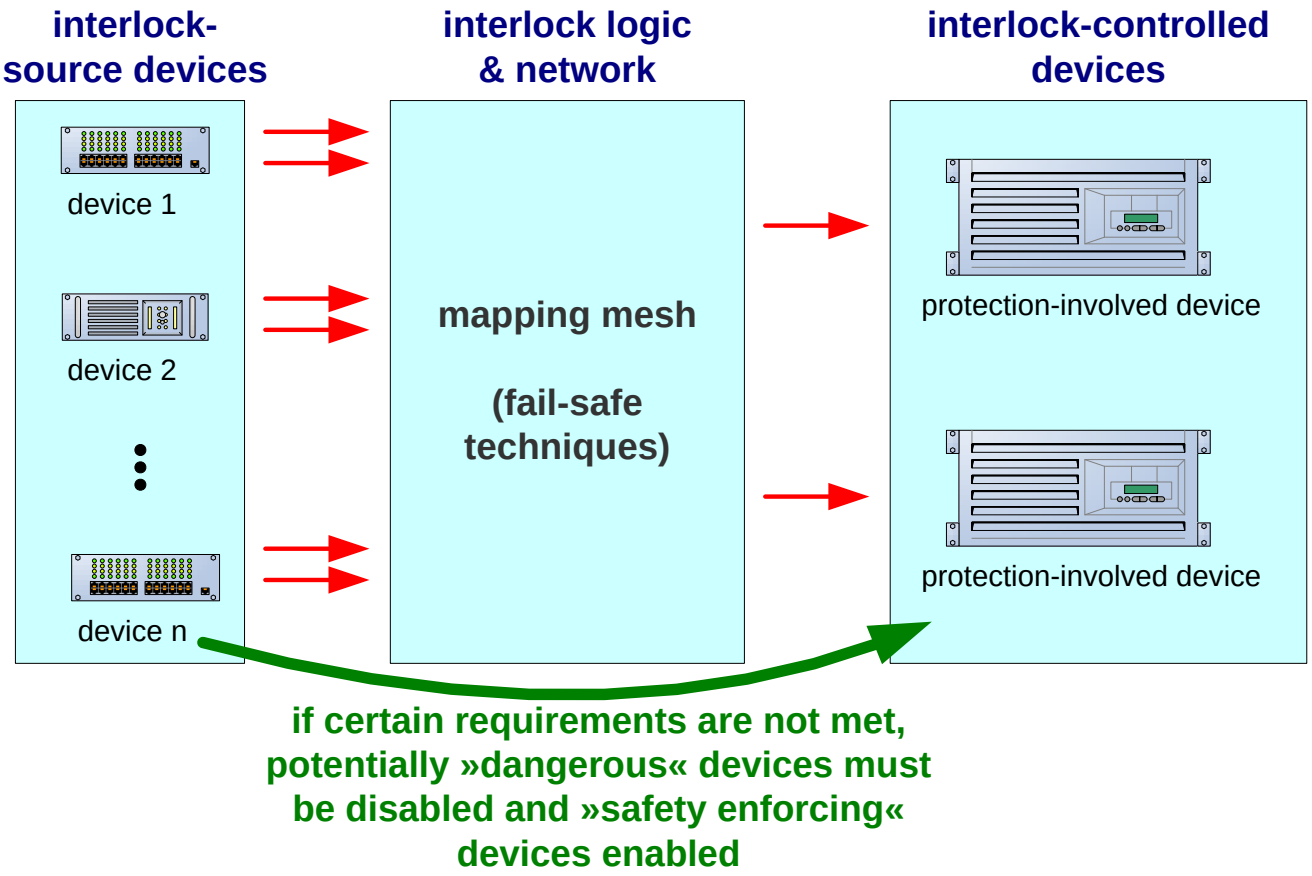
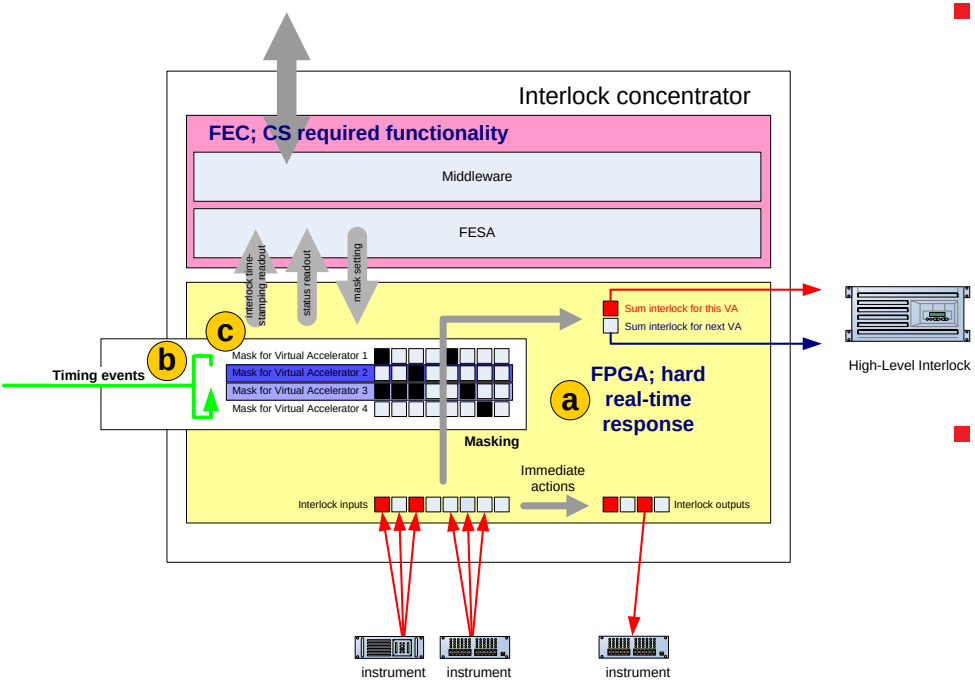


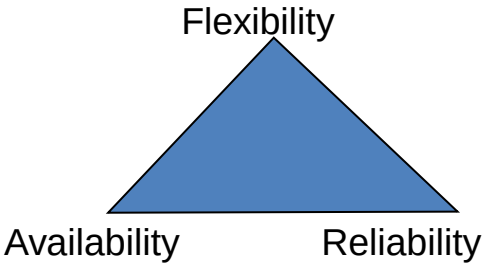
Figure 20: From the protection system point of view there are two types of devices; interlock-source devices and interlock-controlled devices.

Machine Protection and Interlocks 2/2



Interlock concentrator schematic overview

- Balance between



- High Flexibility required: state-specific interlocks
  - Includes complex FPGA logic
- ITER and ESS: interlock system is equally important as the main control system

HARDWARE ISSUES (WHO CARES ABOUT THAT ANYWAY?)

COSYLAB

Hardware platforms

COSYLAB

- VME, cPCI(e), PXI, xTCA (MTCA.4), etc.
- Why don‘t all the labs make the same choice?
- Potential criteria for evaluation:
  - Vendor support, maturity, longevity, maximum transfer rate, topology, form factor, availability, software support, user base, etc.

Hardware platforms

COSYLAB

	VME	ATCA	cPCI
Vendor support	High/Declining	Low/Growing	Medium/Stable
Maturity	High	Medium	High
Longevity	Medium	High	High
Max. transfer rate	VME: 40MB/s VME64: 80MB/s VME64x: 160MB/s VME320: 320MB/s	1Gbps, 10Gbps (Gigabit Ethernet); 250MB/s/lane (PCIe)	PCI: 133MB/s PCIe: 250MB/s/lane (up to 16 lanes)
Topology	Master-slaves	Star Dual star Full mesh	Master-slaves

Hardware platforms

COSYLAB

	VME	ATCA	cPCI
Form factor	6U (64 bit) 3U (32 bit)	12U (ATCA) 2U (µTCA)	3U
High availability	Medium	High	Medium
Software support (Linux, EPICS)	High	Medium	Medium
Cost	High	High	Medium
Users	SNS, SLS, Diamond Light Source, NSLS II, …	XFEL (LLRF), ITER, TPS (considering)	ALBA, TPS, CERN (LHC collimation), LANL, ORNL, ITER (planned)

Hardware platforms

COSYLAB

- Main criteria for selecting hardware platform should be
  - Usability
  - Longevity
  - NOT “top performance” or coolness factor

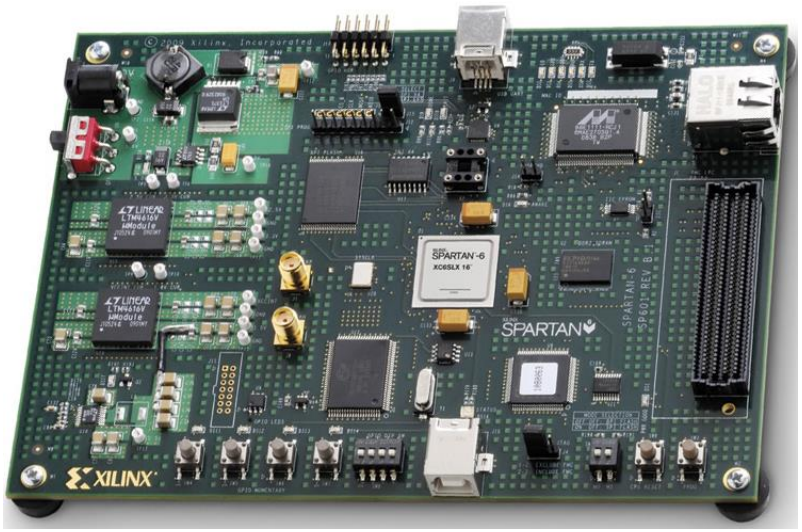
  Acceptance by majority <u>in the industry.</u>

Jean-Francois Gournay (CEA) *: stay with well-improved solutions as much as possible (we use the same analog IOs and binary IOs VME boards - still manufactured - for 20 years.*

Trendy and Sexy – the FPGA

COSYLAB

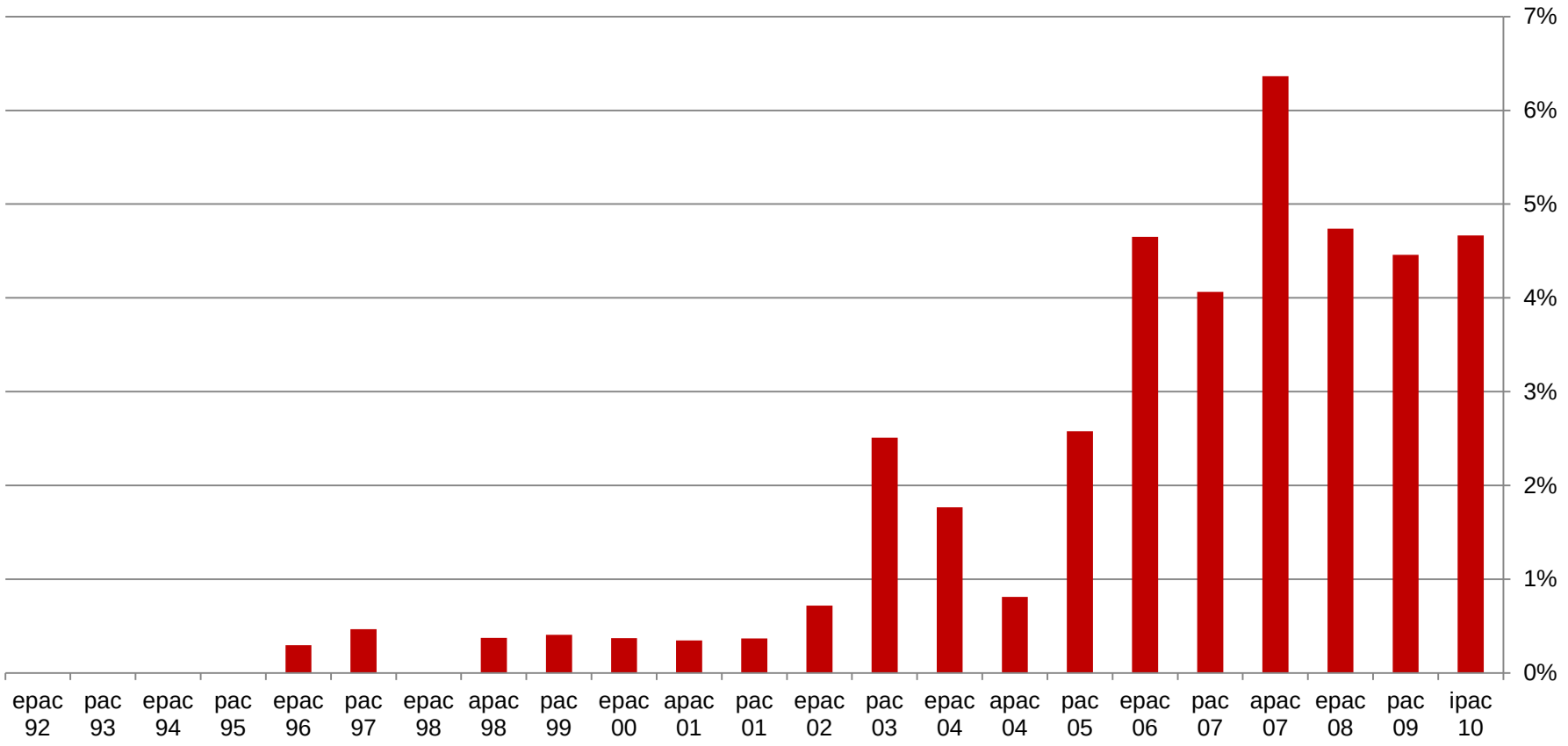
- FPGA: Field Programmable Gate Array
  - Integrated circuit designed to be configured by the customer after manufacturing
    - "reduces hardware development to configuration"
  - Obvious benefits
    - Many inputs and outputs, parallel processing, full synchronization, real time, flexible,…
  - Applications
    - LLRF, MPS, timing, DAQ,…



Trendy and Sexy – the FPGA

COSYLAB

String “FPGA” in conference articles



Mastering complexity requires BRAIN power (more then CPU power)

COSYLAB

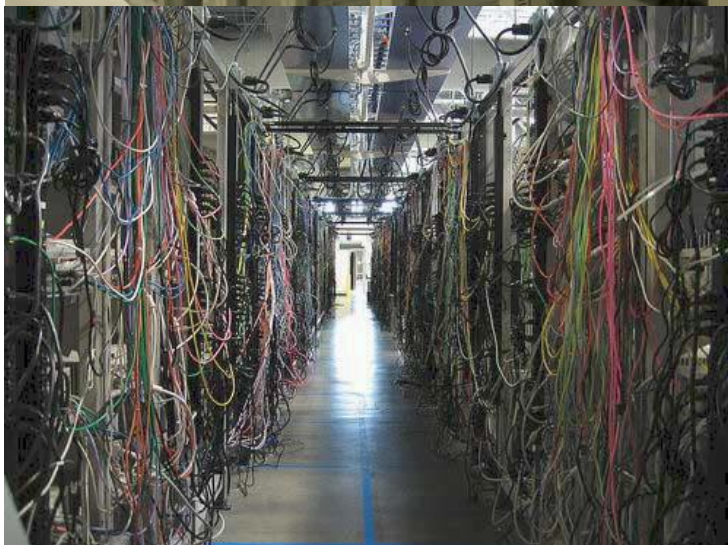
- Advances in technology often give false sense of complexity reduction. Examples
  1. FPGA development environment. For few 100 euros a PC card with free, well supported tools. Feels like an easy start, but it's a marginal win. True challenge is in domain expertise and system knowledge.
  2. System 2.0 syndrome. With new tools and technologies, we'll fix ALL the shortcomings of the system 1.0 … result: a proven, working system is replaced by an over-architected, heavy framework with late delivery.

→ Prudence, use of proven techniques

Elder Mathias (Canadian Light source) : *Be realistic on what is needed to commissioning the machine versus what is needed for optimal performance.*

Who Cares About Cables?

- Always expect trouble! :)
  - Devices not connected properly
  - Cables not wired correctly
  - SW not configured correctly
  - GUIs need to be tweaked
- Users not trained enough (RTFM)
  - Erroneous bug reports
- Integrators must be prepared for all of this



CONTROL SYSTEM DEVELOPMENT

COSYLAB

How To Build A Control System

COSYLAB

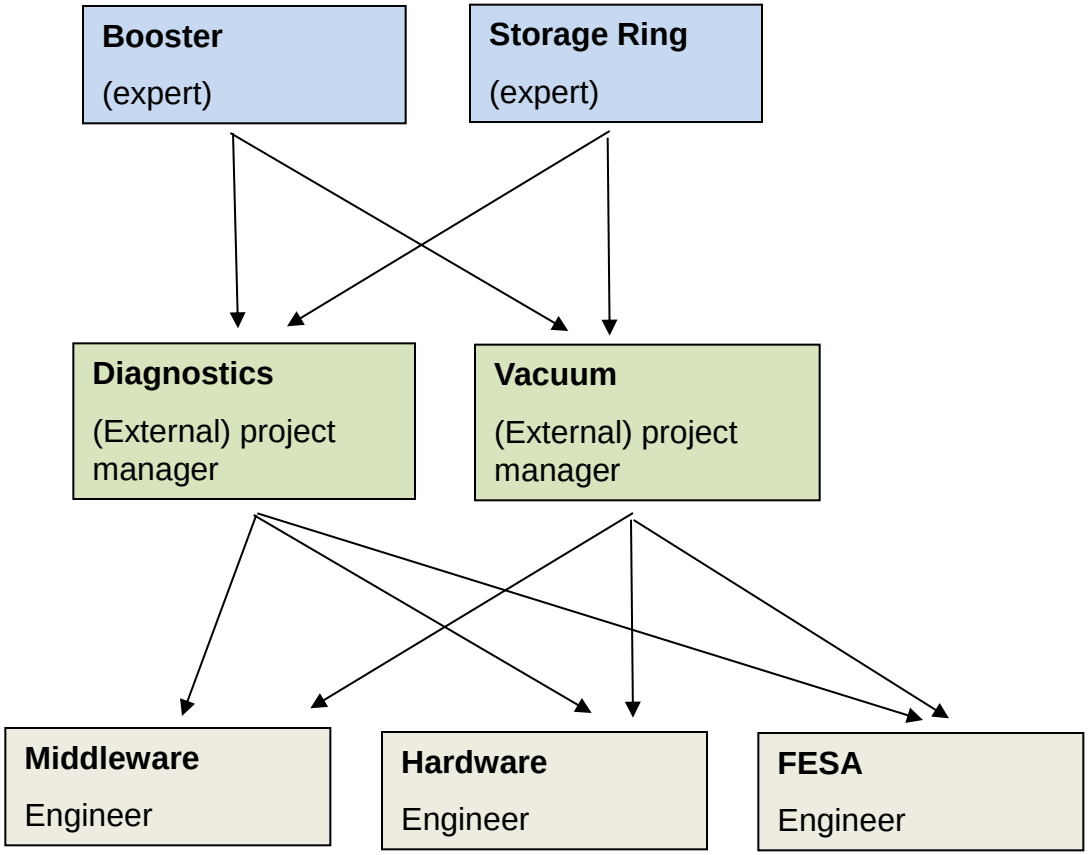
- Control System (CS) is not a DVD with an installation wizard, but rather:
  - Engineering according to specifications
  - Configuration of packages like EPICS, TANGO, FESA, TINE, DOOCS, MADOCA, LabView…
  - Outsourcing or in-house software and hardware development
  - Installation

Role of CS in the project

COSYLAB

- **Relatively low technical risk**
- **Higher organizational risk**
  - Collaboration across all the departments
  - Control system comes late in the project
  - Integrates with most of other subsystems
- Control Systems are an **engineering** discipline like all the others, but with an even more complicated cycle
  - Write specifications
  - Architecture
  - Design
  - Prototyping – **fun part**
  - Test procedures
  - Implementation (coding)
  - Documentation
  - Testing
  - Debugging
  - Acceptance
- Iterative development (evolution through upgrade phases)

Teams and Division of Responsibility



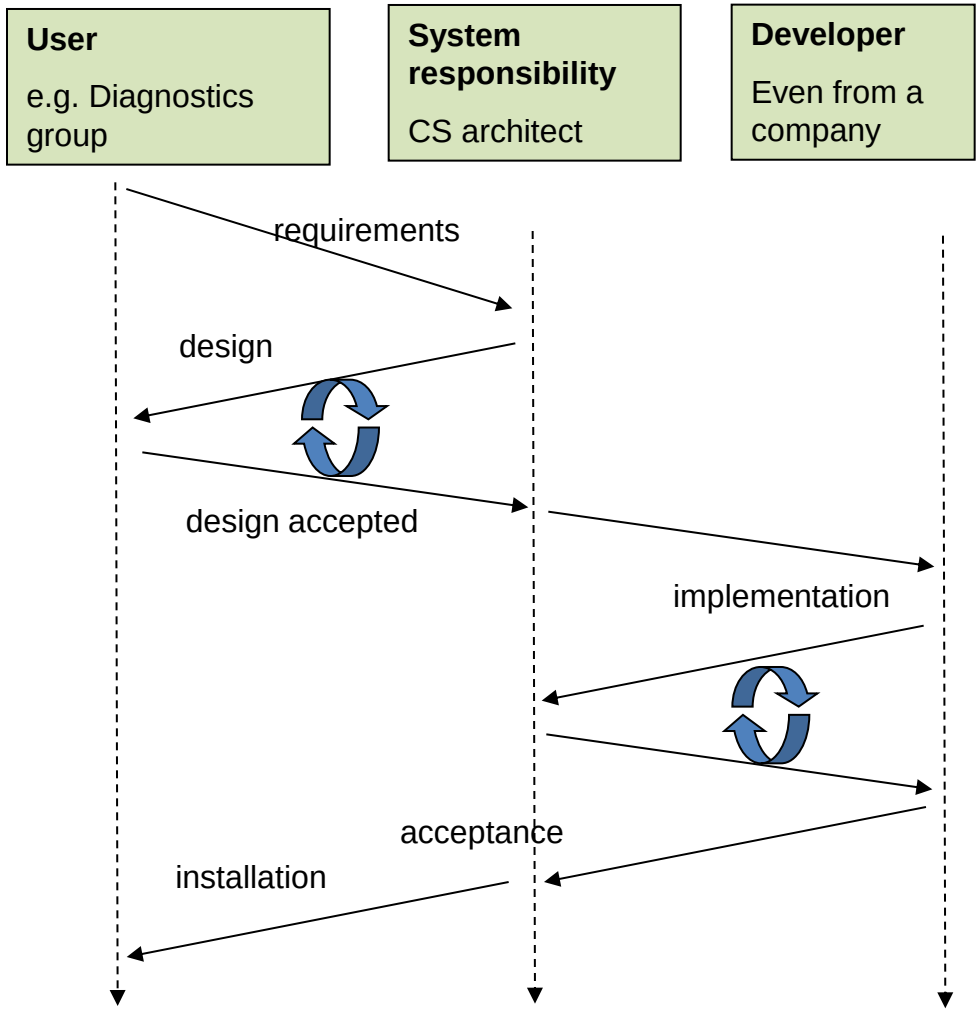
Coordinators / Liasons

Design team (for discussions)

Engineers

Divide Work, but Maintain System Responsibility

- Define development procedures rigorously
- Divide the development among:
  - Within CS group
  - Other groups, experiments,…
  - Outsourcing (companies)
- Use a lightweight project tracking and reporting tool
- CS group defines acceptance procedures



Development process – our experience from 100+ projects

COSYLAB

- Start with requirements very early
  - They will change later in any case, no matter how long you wait for the “final” requirements!

- Standardize development
  - Applies to the whole cycle: design, implementation and testing procedure.
  - More important than standardizing components.

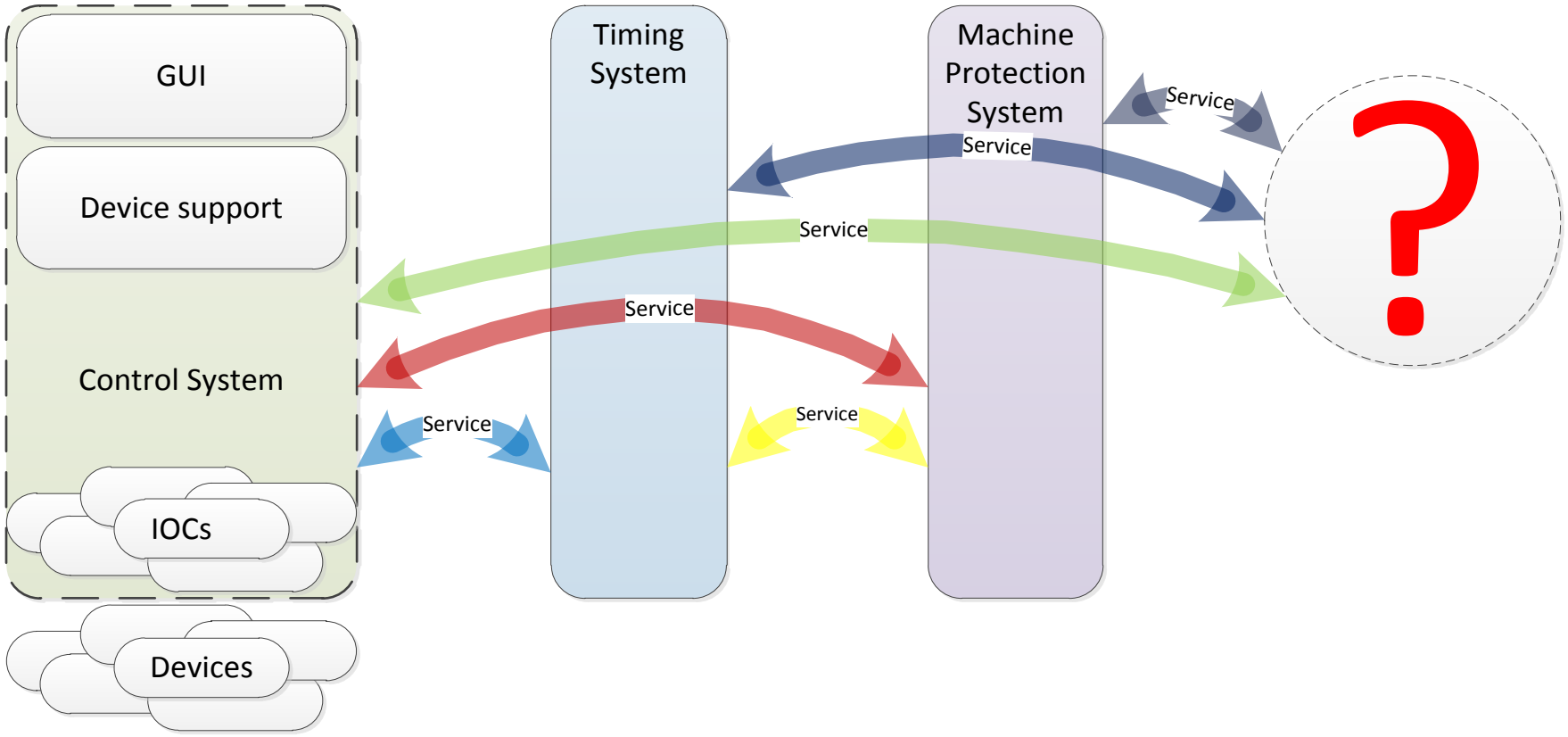
Matt Bickley [JLab]: *The aspect that I think has been most helpful has been standardization of hardware and software[...] Another decision was the choice to develop and rigorously adhere to standard testing and implementation procedures.*

Development process – our experience from 100+ projects

COSYLAB

- Vertical prototypes from the beginning
  - With integrated software and hardware
  - E.g. vertical column (MedAustron), Control Box (ESS), Fair Host Machine (GSI/FAIR),…
- Iterate frequently
  - Yearly cycles
  - First specific requirements usually come when people comment on the first prototype!

Define Systems and Interfaces



- List ALL needed services
- Define systems without duplicating services
- Understand connections between systems
- Define interfaces between systems
- **All must agree on them early on in the project**

FINAL THOUGHTS

COSYLAB

Should we expect clear technological “winners”?

COSYLAB

- Accelerator CS is a very broad field with specific needs for every job
  - Timing needs
  - Safety needs
  - Reliability uptime needs
- Many installations are experimental by nature

- Computing power/$ grows faster then project size
- Increasing expectations in power and flexibility of the CS
- Every added (cheap) CPU increases the entropy of the system

Many arguments for very diverse approaches that blur the overall picture

Increasing challenge of managing the added complexity

Future challenges

COSYLAB

- Large international projects with in-kind contributions: not technical, but managerial challenge.
  - How can CS help
  - How should we design the CS to solve these issues

Conclusion

- CS has shifted over the years
  - From research to engineering
  - From performance to integration challenges
- But architecture and platforms are more or less stable
- So CS development task is how to integrate everything into the CS in-time, on-budget, and with a low-risk by using an increasingly large number of off-the-shelf components.

THANK YOU!

Mark Plesko, mark.plesko@cosylab.com

COSYLAB

Tel.: +386 41 934550

Web: www.cosylab.com