Elke Zimoch :: Section Controls :: Paul Scherrer Institut

Accelerator Controls

JUAS 2020

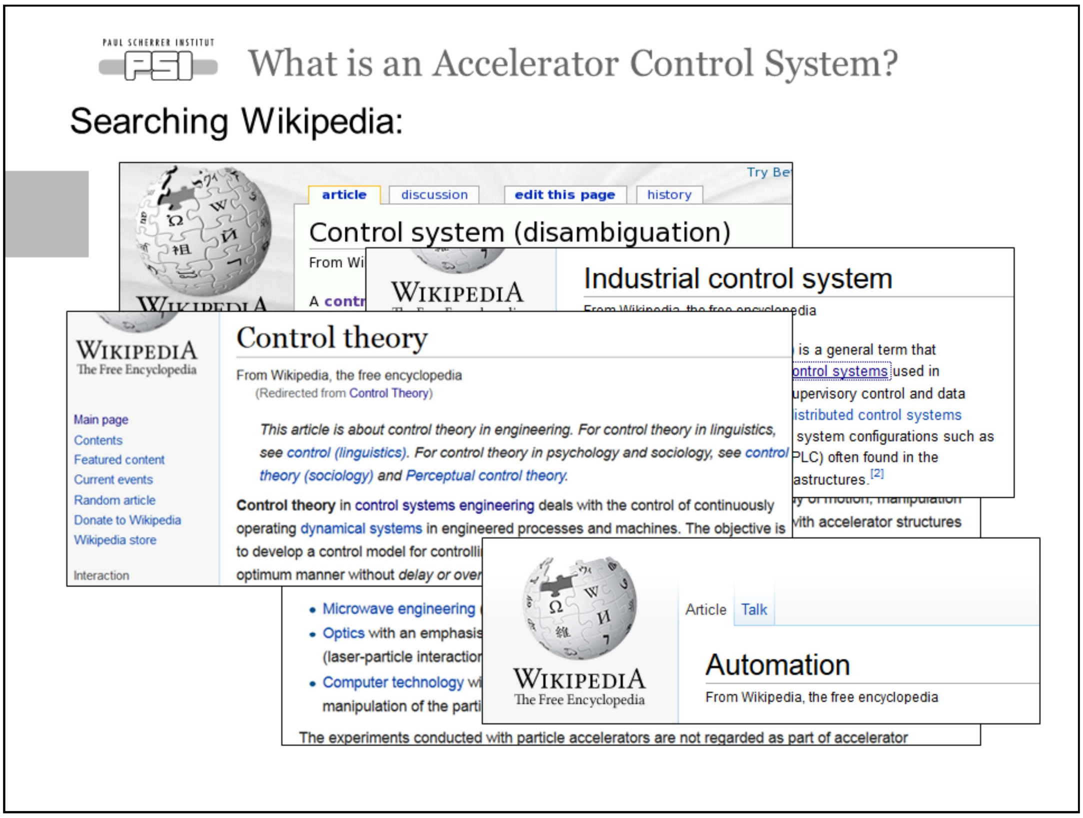
Why talking about Accelerator Controls?

Soon in the future (and once upon a time):
Scientist **Dr. Example Guy** wants to do
VeryImportantMeasurement_OneDotOne
for that he creates some actuators and detectors
Super_Creative_HardwareSolution
puts it into the accelerator
and calls the Controls Group
"Please make it run".

I want to teach you a minimum awareness about the control system that «runs» the accelerator ...

PAUL SCHERRER INSTITUT PSI

Table of Content



- From Wikipedia (https://en.wikipedia.org/wiki/Main_Page):
- **Control System:** A control system is a device or set of devices to manage, command, direct or regulate the behavior of other devices or systems.
- **Accelerator Physics:** Accelerator physics is a branch of applied physics, […]. It is also related to other fields: […] Computer technology with an emphasis on digital signal processing; e.g., for automated manipulation of the particle beam.
- **Industrial Control System:** Industrial control system (ICS) is a general term that encompasses several types of control systems used in industrial production, […].

- **Control Theory:** Control theory in control systems engineering deals with the control of continuously operating dynamical systems in engineered processes and machines. The objective is to develop a control model for controlling such systems using a control action in an optimum manner without delay or overshoot and ensuring control stability.
- **Control Systems Engineering:** Control engineering or control systems engineering is an engineering discipline that applies automatic control theory to design systems with desired behaviors in control environments. The discipline of controls overlaps and is usually taught along with electrical engineering at many institutions around the world
- **Automation:** Automation or automatic control, is the use of various control systems for operating equipment such as machinery, processes in factories, […] and other applications with minimal or reduced human intervention.

- Unfortunately the „Controls Theory“ does not cover highly complex and diverse systems like accelerators. So, there is no theory to learn.
- Conclusion: This is all related to Accelerator Control Systems, but does not hit the point.

- Best match found on Wikipedia is the article about **Distributed Control Systems** https://en.wikipedia.org/wiki/Distributed_control_system

What ~~is~~ does an Accelerator Controls System (1/6)

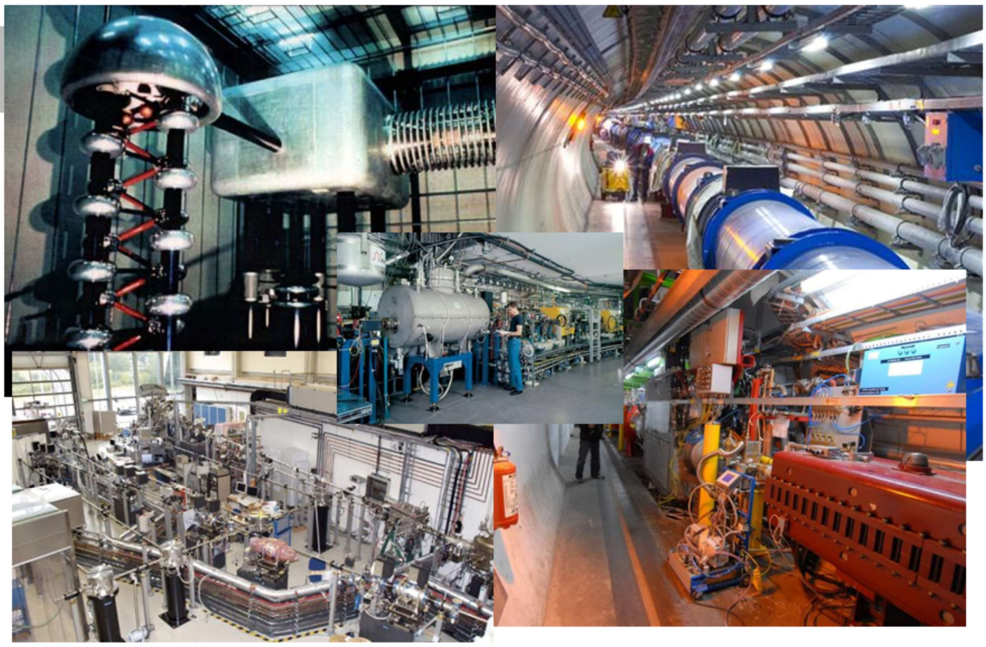
- Controls the accelerator (Source, Magnets, RF)
- Provides diagnostics information (BPMs, Cameras)
- Monitors environment (Vacuum, Temperature)
- Feedback programs for beam parameters (orbit feedback)
- Makes "the machine" running and controllable …

… reliable, with good performance, flexible … economical safe (without producing black holes and destroying the world)

- As there is no easy answer to what an Accelerator Control System is, lets try to find out what it does.

PAUL SCHERRER INSTITUT PSI What does an Accelerator Control System? (2/6)

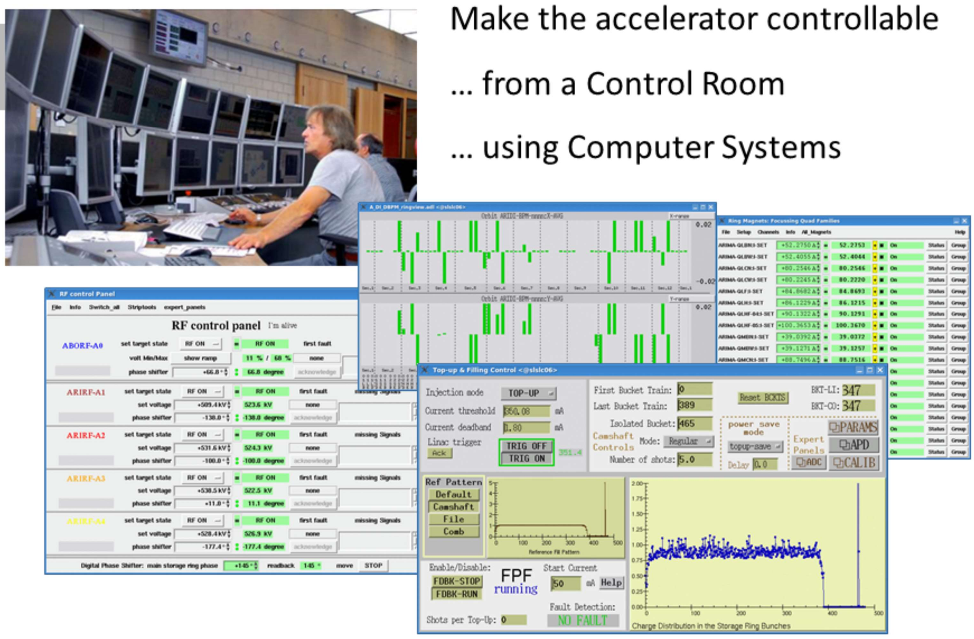
Controls the accelerator hardware:



- **What does an Accelerator Control System actually control?**

- For example the number of components needed to run an accelerator:
- Swiss Light Source (SLS at PSI) Accelerator components to control:
  - ca. 200 computers
  - ca. 600 magnets (+power supplies)
  - 300 vacuum pumps
  - 9 cavity structures
  - ca. 150 beam position monitors
- 21 beamlines (together)
  - ca. 300 computers
  - 10 undulator magnets
  - more than 1200 motors
- Distances between components (stroage ring with 130 m diameter): 50 km power cable and more than 500 km signal cable

What does an Accelerator Control System? (3/6)



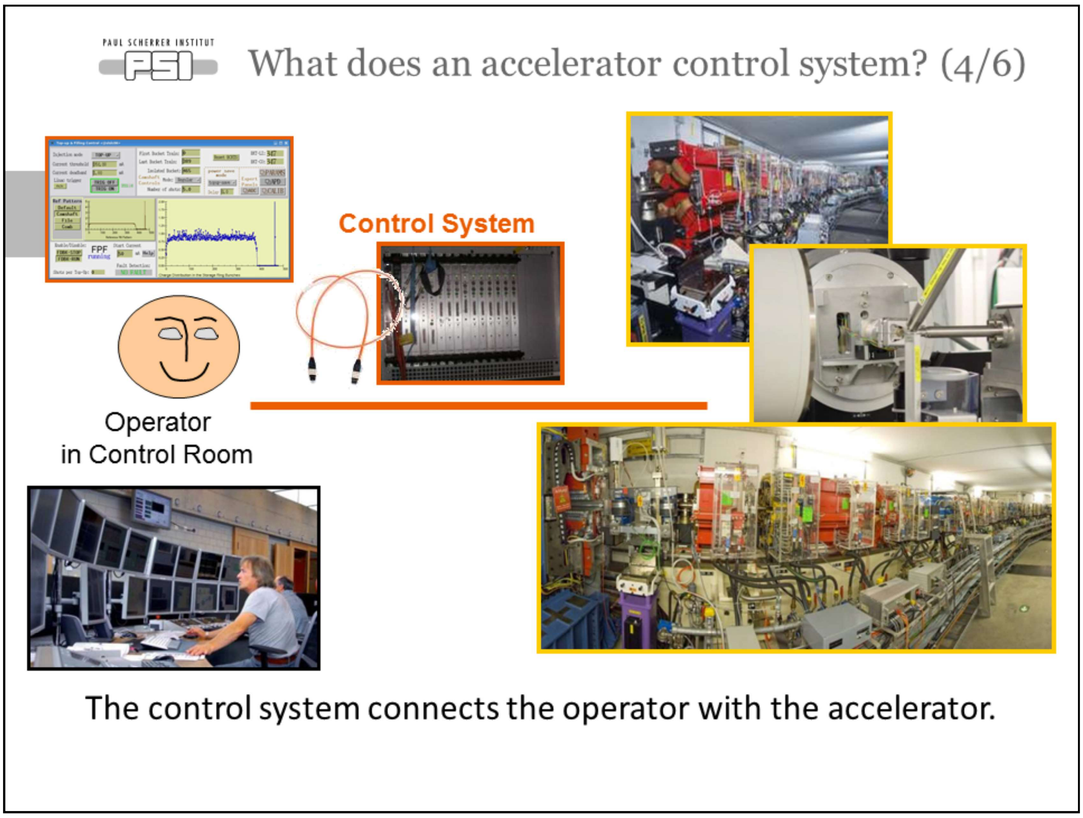
Make the accelerator controllable

… from a Control Room

… using Computer Systems

- **For what do you need an Accelerator Control System (continued)?**

- Some examples of applications and programs needed to run an accelerator:
- Device Control Panels
- Feedback Systems (e.g. orbit feedback, filling pattern feedback)
- Archiving and Data Acquisition Systems (Storage)
- Scan tools for experiments and measurements
- Conditioning tools for RF components
- Simulations and comparisons with real accelerator
- Alarm handling and machine protection



- The Accelerator Control System connects the Operator in the control room with the accelerator hardware. The control room might not be near the accelerator:
- SLS is 200m away from PSI main control room
- For SwissFEL the control room will be a kilometer away
- Some experiments are controlled remote from all over the world

What does an Accelerator Control System? (5/6)

Accelerator Control System

Inputs

- Values from all diagnostic devices (BPMs, screens, ICT/PCT, etc.)
- Information from the experiment
- Read back values from devices (Magnets, RF, etc.)
- Read back values from environment (Vacuum, Temperature, etc.)

Operator

Outputs

- Set values from devices (Magnets, RF, etc.)
- Actuators of the environment (vacuum pumps, heaters, chillers, etc.)
- Triggers
- Alerts (SMS, emails, alarms, etc.)

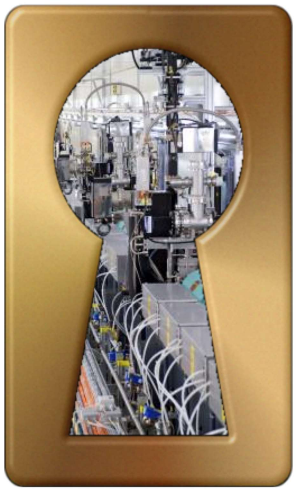
What does an Accelerator Control System? (6/6)

The

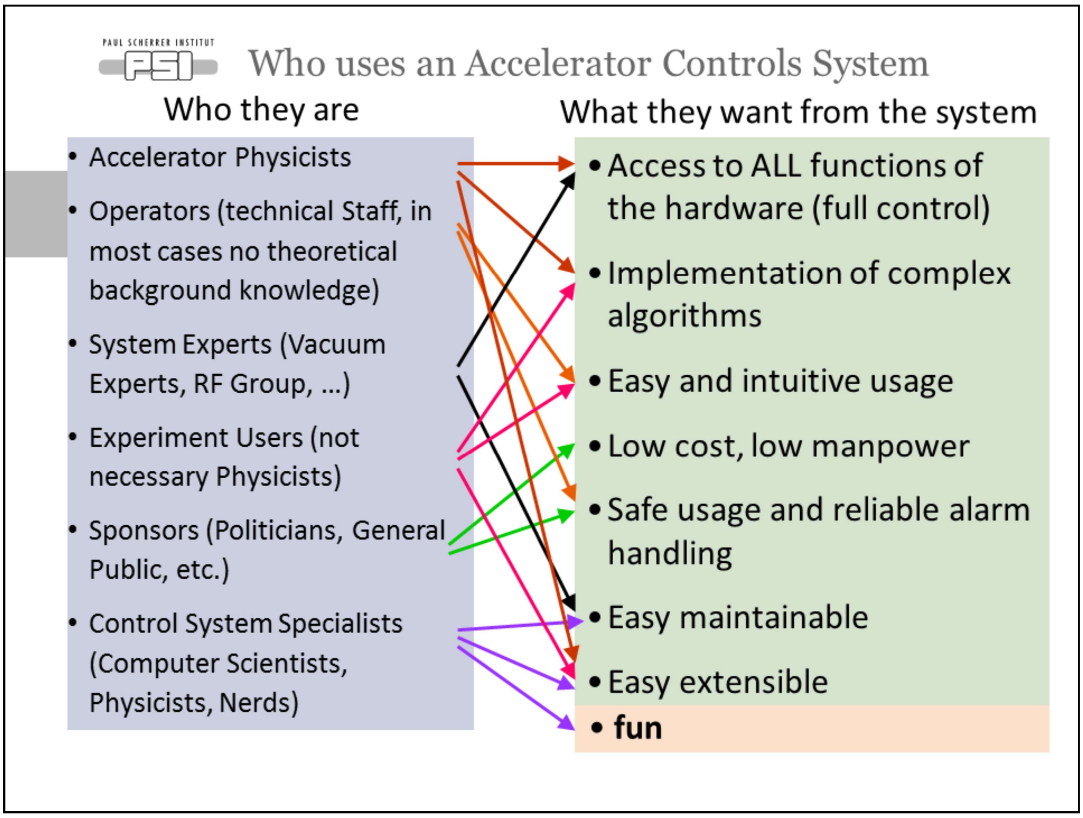
Accelerator Control System

- does provide a keyhole view on the accelerator

- is the only way to access any component remotely



- **Conclusion:**
- The Accelerator Control System is the only connection between the Control Room (Operator) and the real hardware.
- Everything that is not shown by the control system can not be seen.
- If the control system is the only way to access hardware, all systems that are needed for this remote access belong to the control system.



- The accelerator control system is used by many different groups for many different purposes – and has to support them all.

What is the Technical Environment?

Control Systems (one way or another) have to deal with …

- **Distributed** end points and processes
- **Data Acquisition** (front end hardware)
- **Real-time** needs (where necessary)
- **Process control** (automation, feedback, PID controller)
- **Central Services** (Archive, Databases, Name Resolution)
- **Data transport** (control system protocol, network)
- **Security** (who's allowed to do what from where?)
- **Time synchronization** (time stamps, cycle ids, etc.)

that is:

Computers (in different flavors) and **Computer Environment**

PAUL SCHERRER INSTITUT

What is an Accelerator Controls System

Definition:

An **Accelerator Control System** is a computer environment that allows remote access to the accelerator hardware with a lot of different functionality to satisfy the requirements of several different user groups.

In addition a modern

Accelerator Control System:

tries to unify the access to different

hardware

(one way to rule them all)

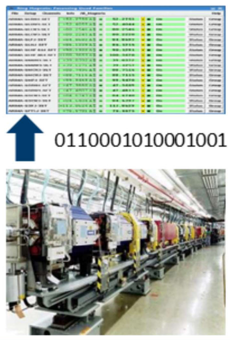
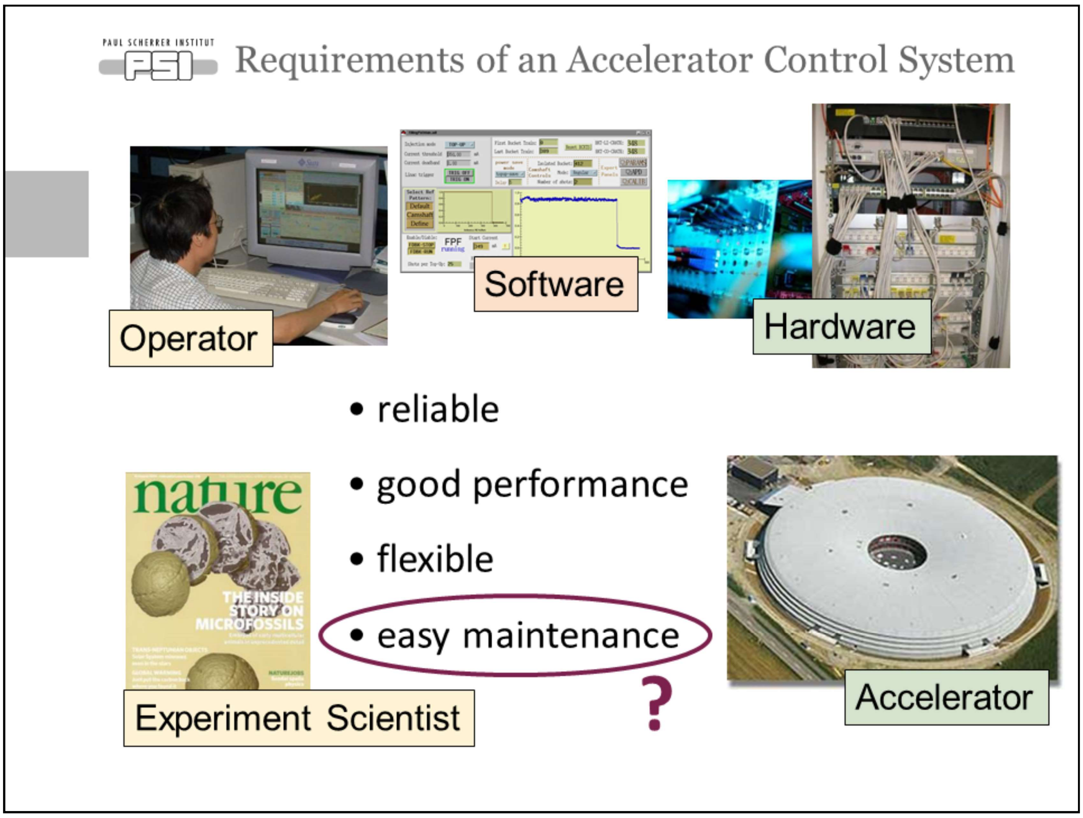


Table of Content

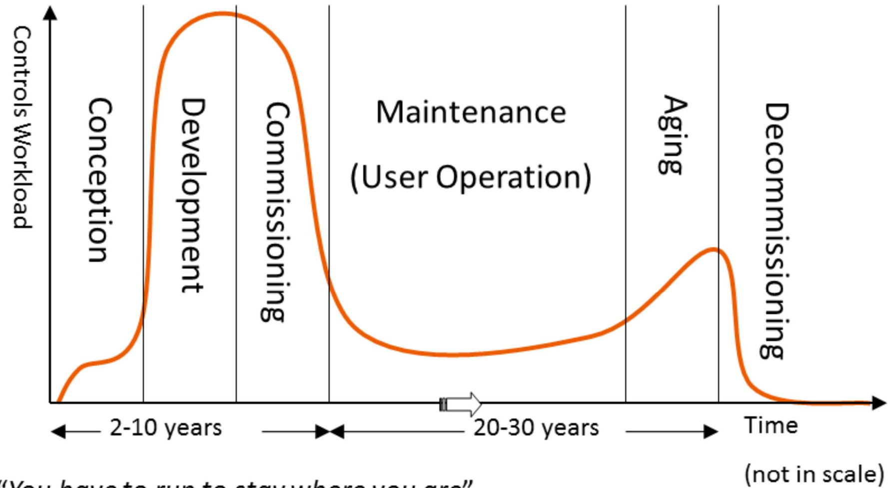


- The requirements from different user groups might be different.

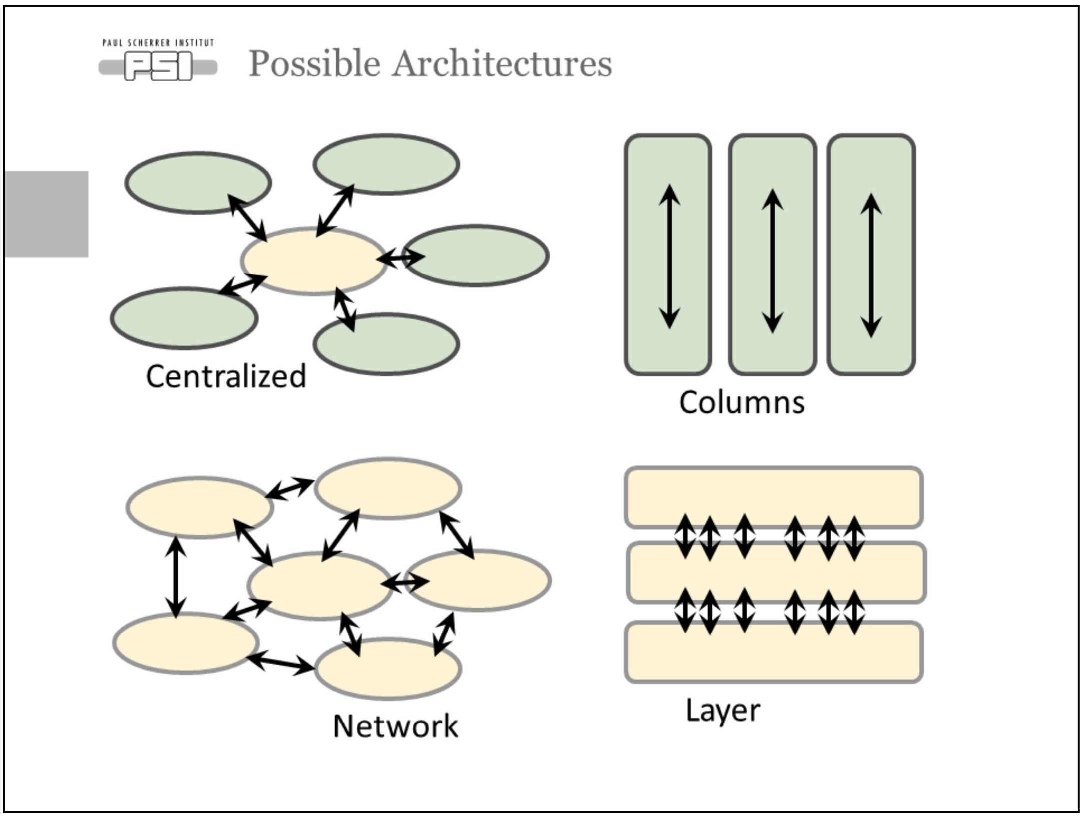
- The most common requirements are:
- Reliability – if I have beam time scheduled, I want it to happen
- Good Performance – speed, responsiveness, modern possibilities
- Flexibility – „hey, I have a good idea, lets try …“
- Easy maintenance – this comes with limited resources

Why is easy Maintenance important?

Controls System Lifecycle:



- *"You have to run to stay where you are"*
- Workload never got to zero during accelerator lifetime
- Normal accelerator lifetime ~ 30 to 40 years



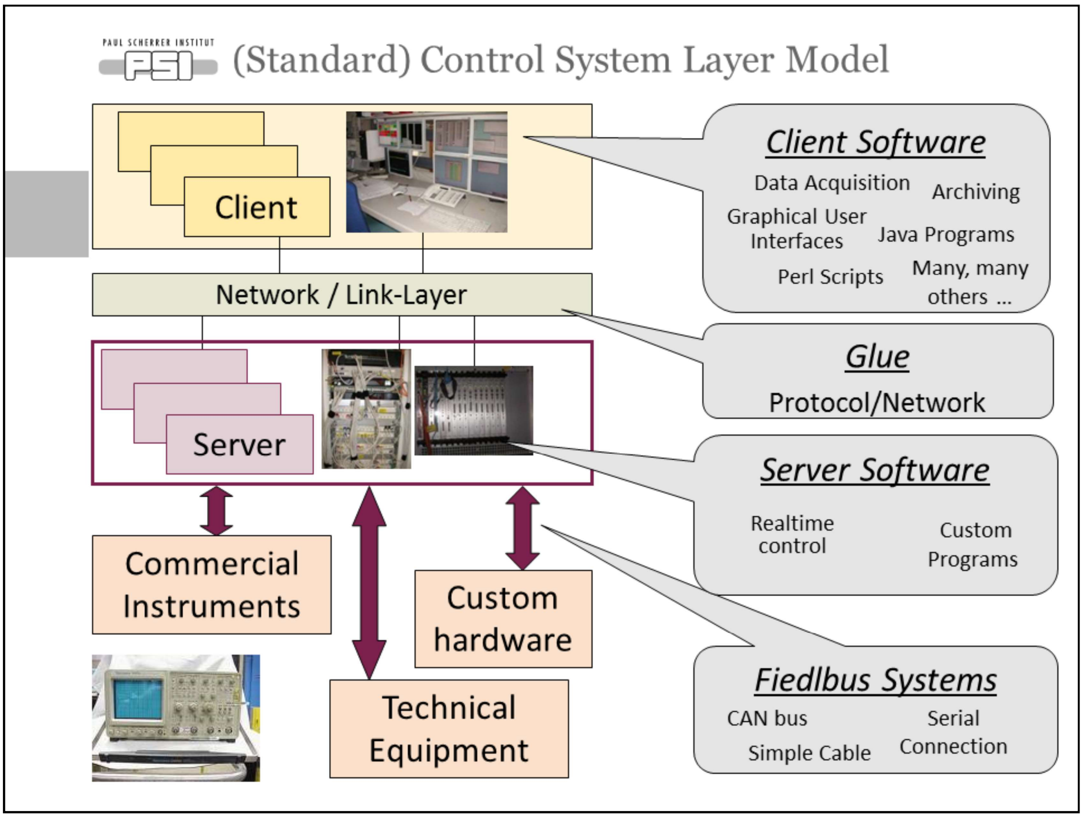
- **Centralized:**
- Single point of failure.

- **Columns:**
- No exchange of information between different systems.

- **Network (Peer-to-peer):**
- Can be difficult to maintain if an addition has to be added to all other systems.

- **Layer:**
- Can be difficult to find out where a information came from.

- **Practical solution:**
- Mix between some or all of those architectures.



- Three-tier standard model for distributed control systems.

- The hardware layer was considered dumb when the terminology was developed.
- Meanwhile „hardware" can be as complicated as a robot (with its own controller) or an oszilloscope (with an operating system installed on a specialized computer).
- More and more logic is moved to the hardware layer.
- Therefore, I would like to call it now a 4-tier model.

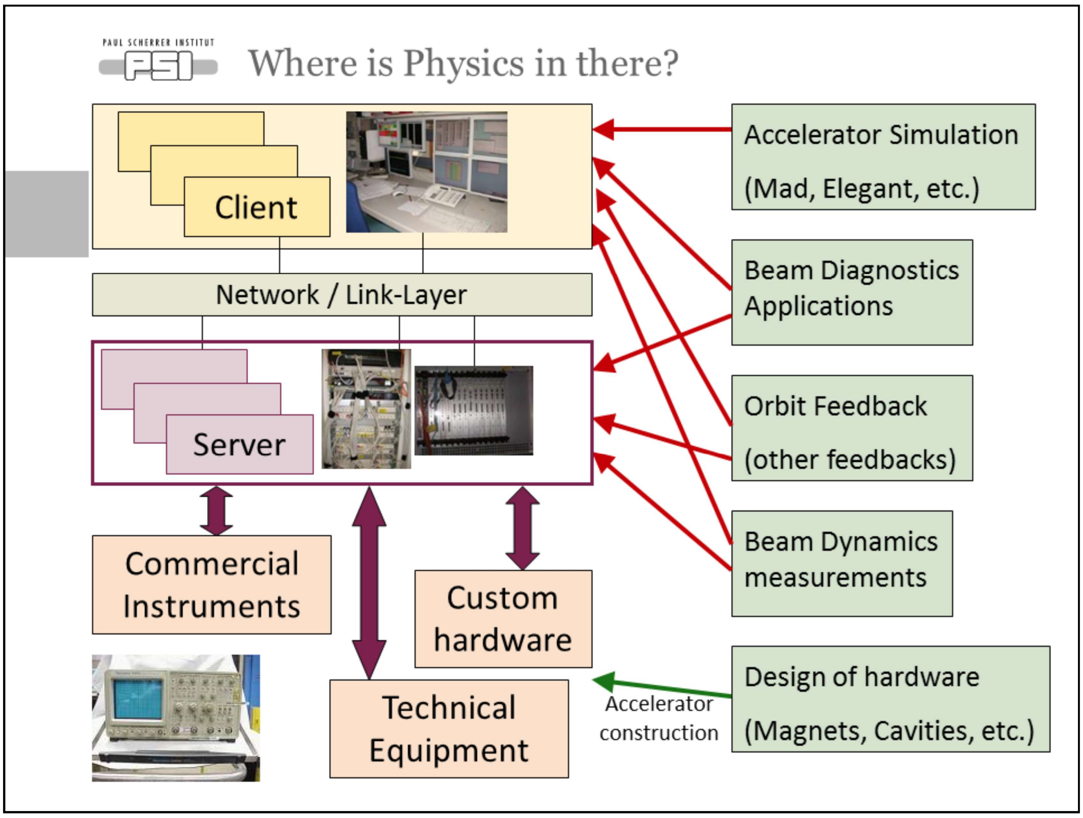
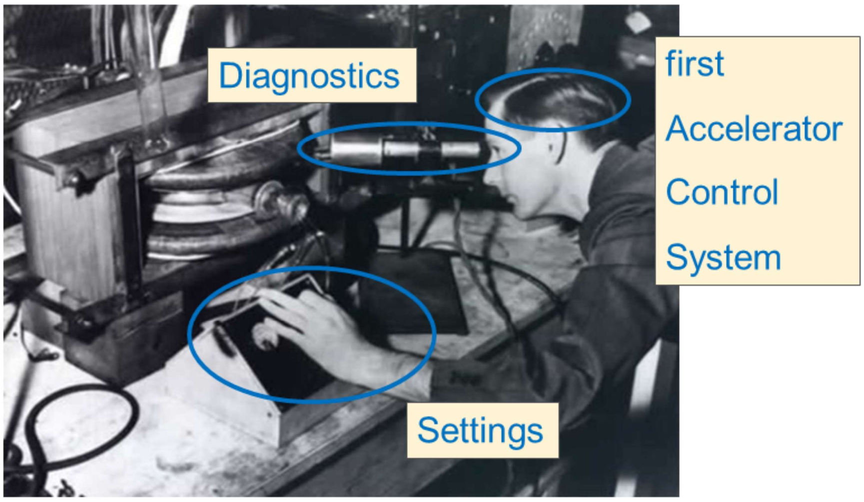

PAUL SCHERRER INSTITUT
PSI
Where is Physics in there?
Client
Network / Link-Layer
Server
Commercial Instruments
Technical Equipment
Custom hardware
Accelerator Simulation
(Mad, Elegant, etc.)
Beam Diagnostics Applications
Orbit Feedback
(other feedbacks)
Beam Dynamics measurements
Design of hardware
(Magnets, Cavities, etc.)
Accelerator construction

Table of Content

History of Accelerator Controls (1/3)



Donald Kerst with the first betatron, invented at the University of Illinois (USA) in 1940

- https://physics.illinois.edu/people/history/betatron.asp

- No Control System needed because:
- Direct tuning with knobs
- Direct diagnostics with own eyes
- (Very) limited number of devices involved

- > so the brain is the Accelerator Control System in this case

PAUL SCHERRER INSTITUT PSI

History of Accelerator Controls (2/3)

- High Left: AGS control room, circa 1966
- Alternating Gradient Synchrotron (AGS) is a Proton Accelerator at Brookhaven National Labs (BLN) in Long Island (USA)

- Lower Right: Cern Control Room 1974
- Jean-Pierre Potier (turning buttons) and Bertran Frammery (telephoning) on shift. The 26 GeV Synchrotron and later also its related machines (Linacs 1,2,3; PS-Booster; LEP-Injector Linacs and Electron-Positron Accumulator; Antiproton Accumulator, Antiproton Collector, Low Energy Antiproton Ring and more recently Antiproton Decelerator) were all controlled from the PS control room situated on the Meyrin site. The SPS and LEP were controlled from a separat control centre on the Prevessin site. In 2005 all controls were transferred to the Prevessin centre.
- *Date*: Feb 1974
  Original ref.: CERN-PHOTO-7402124X

History of Accelerator Controls (3/3)

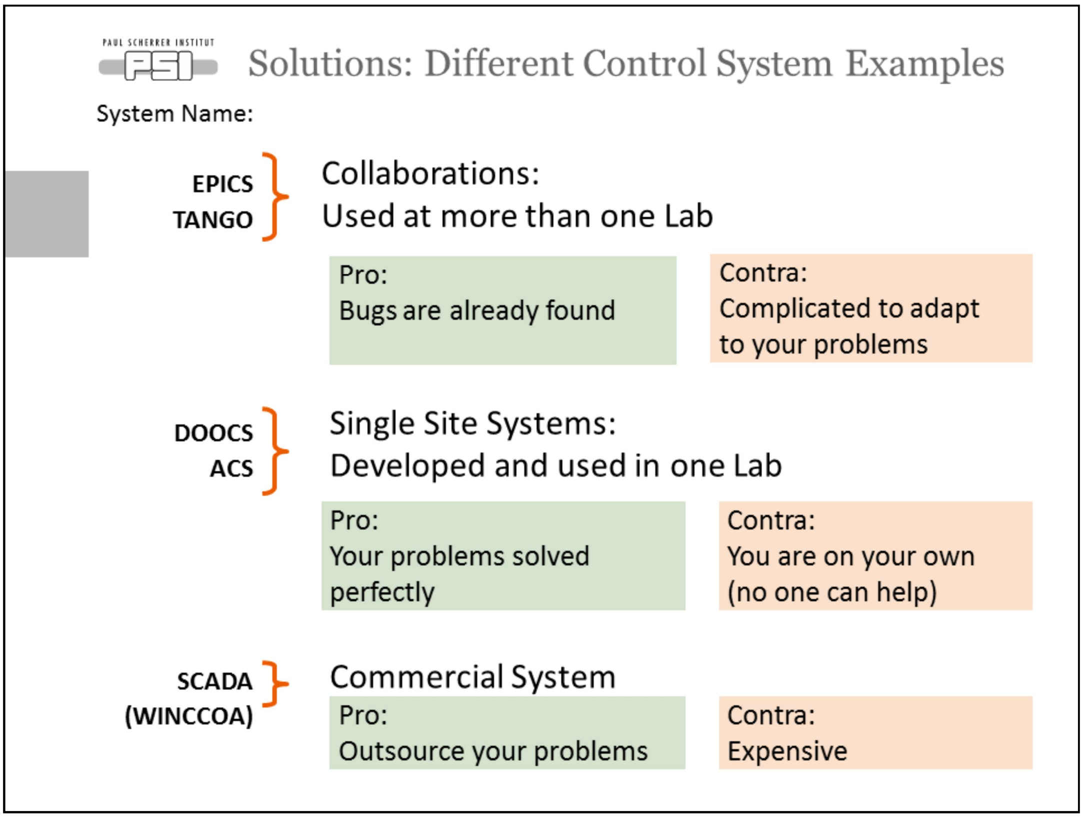
International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS)
First held in 1987 in Villars-sur-Ollon (Switzerland), hosted by CERN.

The term "Control Systems" in ICALEPCS is broadly interpreted to include:

- all components or functions, such as processors, interfaces, field-busses, networks, human interfaces, system and application software, algorithms, architectures, databases, etc.
- all aspects of these components, including engineering, execution methodologies, project management, costs, etc.

https://www.icalepcs.org/ - Next ICALEPCS will be held 5-11, October, 2019 in New York, USA

- Exerpt from the dinner speech of Roland Müller on ICALEPCS 2013 (https://www.icalepcs.org/uploads/documents/LAA-speeches-2013.pdf):
- […]
- In the 1980ˊs control system has not been seen as an essential part of accelerator projects. No forum existed to share expertise. On the other hand control systems have been frequently blamed for project delays and cost overruns.
- Following 2 precursor events in Berlin and Brookhaven 1985 Peter Clout initiated a workshop in Los Alamos, the first devoted entirely to accelerator control systems. 130 participants presented 50 papers eventually published in Nuclear Instruments and Methods.
- 2 years later the follow up event the Europhysics conference COCONF in Villars-sur-Ollon already extended the scope to Large Experimental Physics Control Systems.
- As the very first representative of an Asian institute Shin-Ichi Kurokawa from KEK participated in both events. On his initiative and with strong support from Axel Daneels from CERN it was already then decided to involve Asia as well as Europe and America in all follow-up events.
- The second conference was held at Triumf in Vancouver in Canada in 1989 where the name has been coined. The third followed 1991 at KEK, Tsukuba, Japan, attended by 240 participants. […]



- SCADA = Supervisory control and data acquisition
- WINCCOA is the successor of PVSS II from Simens.

- Best solution: all systems are used somewhere to control accelerators successful. All systems support good experiments and produce excellent science. Therefore, the choice of system is political not technical.

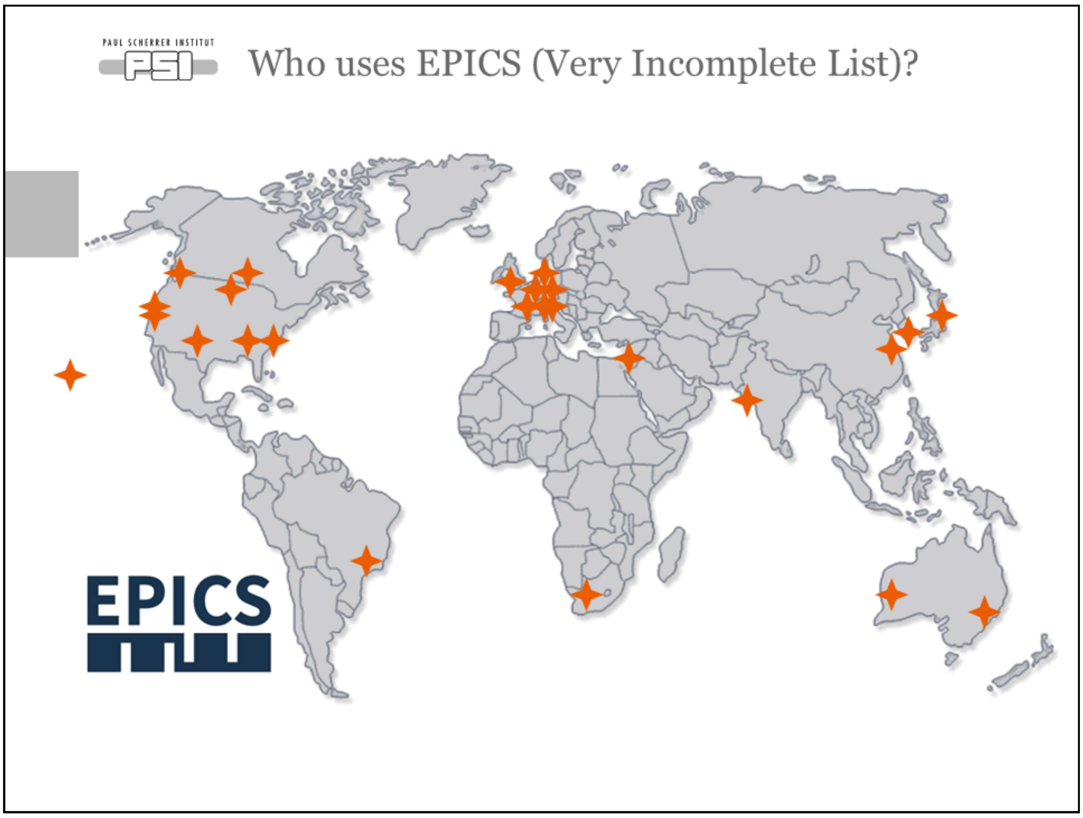
What is EPICS?

- **EPICS** (**E**xperimental **P**hysics and **I**ndustrial **C**ontrol **S**ystem)
  - –is a set of software tools and applications
  - –supports distributed control systems for large research facilities like accelerators
  - –uses Client/Server and Publish/Subscribe methods
  - –uses the Channel Access (CA) network protocol
- In 1989 started a collaboration between Los Alamos National Laboratory (GTA) and Argonne National Laboratory (APS) (Jeff Hill, Bob Dalesio & Marty Kraimer)
- More than 150 licenses agreements were signed, before EPICS became Open Source in 2004

GTA: Ground Test Accelerator
APS: Advanced Photon Source

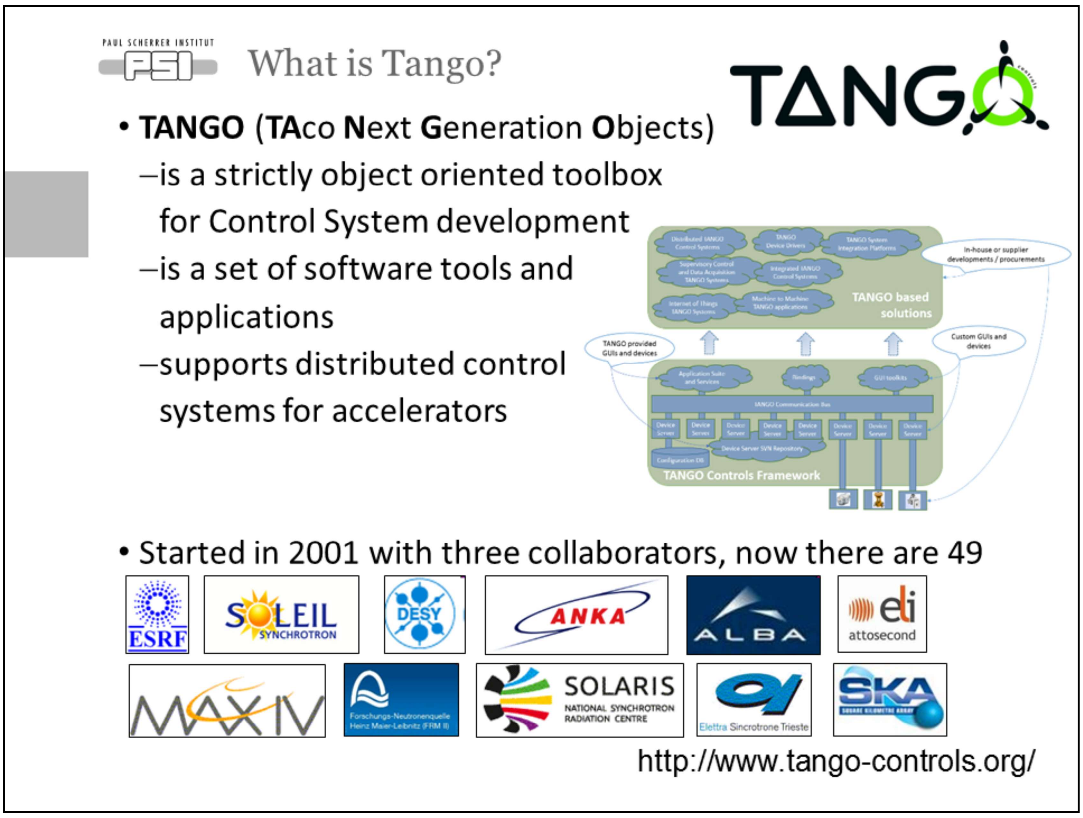
https://epics-controls.org/

- Quote from the EPICS web page (https://epics-controls.org/about-epics/):
- **EPICS** is a set of software tools and applications which provide a software infrastructure for use in building distributed control systems to operate devices such as Particle Accelerators, Large Experiments and major Telescopes. Such distributed control systems typically comprise tens or even hundreds of computers, networked together to allow communication between them and to provide control and feedback of the various parts of the device from a central control room, or even remotely over the internet.
- **EPICS** uses Client/Server and Publish/Subscribe techniques to communicate between the various computers. Most servers (called Input/Output Controllers or IOCs) perform real-world I/O and local control tasks, and publish this information to clients using robust, EPICS specific network protocols Channel Access and pvAccess. These protocols are designed for high bandwidth, soft real-time networking applications that EPICS is used for, and is one reason why it can be used to build a control system comprising hundreds of computers.



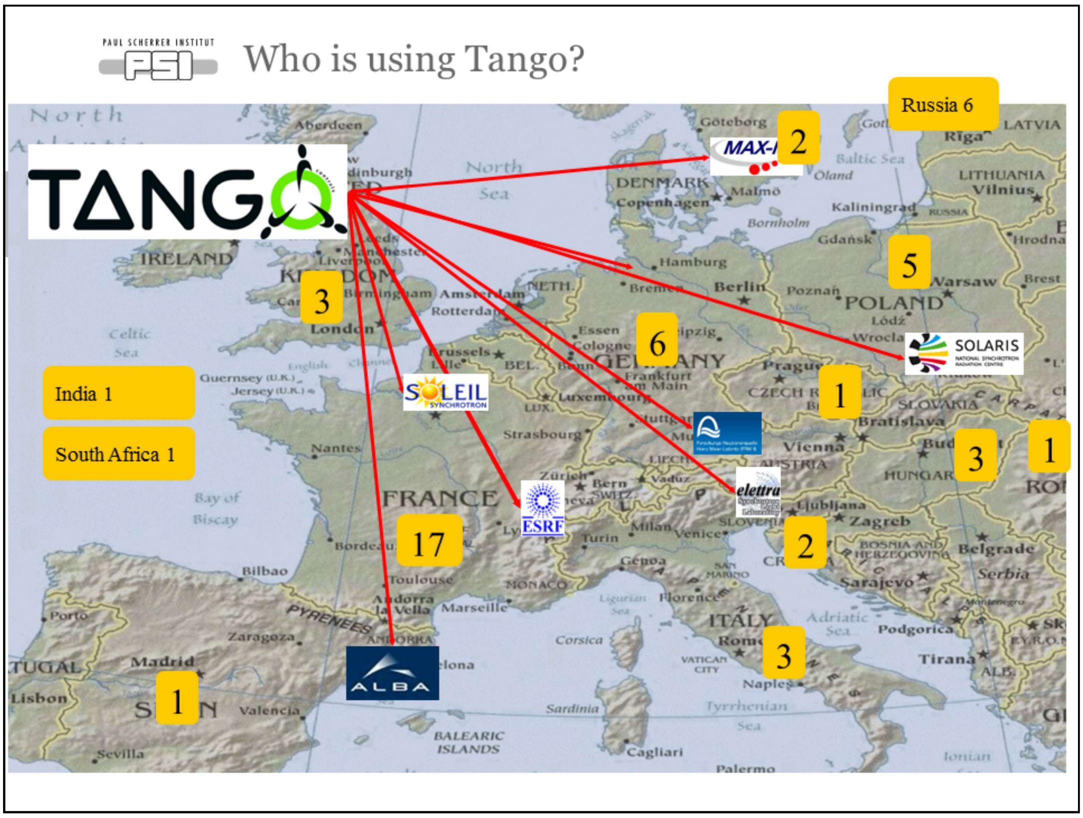
- Some EPICS users (very incomplete list):

- The Advanced Photon Source at Argonne National Laboratory
- Australian Synchrotron
- Berlin Electron Synchrotron (BESSY II)
- Brazilian Synchrotron Light Source (LNLS)
- Deutches Elektronen Synchrotron (DESY)
- Diamond Light Source
- Fermilab (FNAL)
- Jefferson Laboratory (JLAB)
- Keck Observatory
- KEK B-Factory
- Laboratori Nazionali di Legnaro (INFN-LNL)
- Lawrence Berkeley National Laboratory (LBL)
- Los Alamos National Laboratory (LANL)
- Swiss Light Source (SLS/PSI)
- Spallation Neutron Source (SNS)
- Stanford Linear Accellerator Center (SLAC)



- http://tango-controls.readthedocs.io/en/latest/overview/overview.html#what-is-tango-controls

- CORBA and ZMQ to communicate between device server and clients
- C++, Python and Java as reference programming languages
- Linux and Windows as operating systems
- Modern object oriented design patterns
- Naturally implements a microservices architecture
- Unit tested, continuous integration enabled
- Hosted on Github (https://github.com/tango-controls)
- Extensive documentation + tools, large community

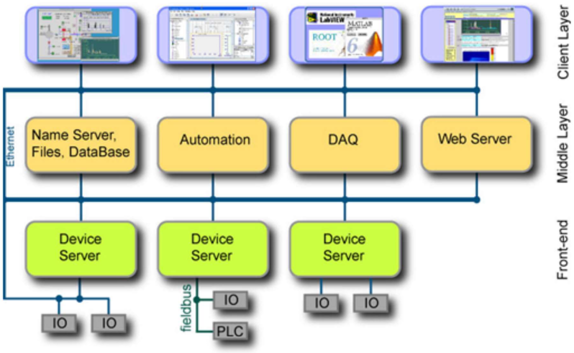


Find out about the 49 community members on http://www.tango-controls.org/partners/institutions/

What is DOOCS (at DESY)?

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

- strictly object oriented system design (C++ and Java)
- Class libraries as building blocks



- Build for FLASH, now used for European XFEL

https://doocs-web.desy.de/index.html

- Quotations from https://doocs-web.desy.de/index.html#about

- The Distributed Object-Oriented Control System - DOOCS - provides a versatile software framework for creating accelerator-based control system applications. These can range from monitoring simple temperature sensors up to high-level controls and feedbacks of beam parameters as required for complex accelerator operations.
- DOOCS is based on an distributed client-server architecture combined with a device-oriented view. The devices are the basic entities and can be virtual i.e. implemented in software or real ones with hardware attached. Each control system parameter is made accessible via network calls through a device application.
- Its transportation layer is based on the standardized, industrial RPC protocol and allows for a robust and efficient data transfer. Support for integrating a timing system providing clock, trigger and other time-based accelerator information is built-in into the core software and comfortably accessible within the framework.

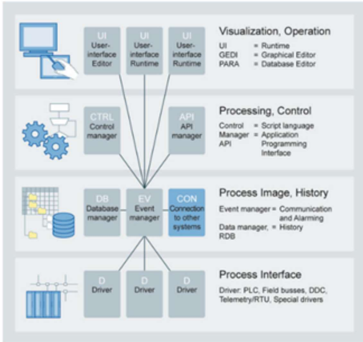
What is PVSS now WinCC-OA (at CERN)?

CERN

WinCC Open Architecture

former **PVSS II** (**P**rozess**v**isualisierungs- und **S**teuerungs**s**ystem 2)

– is an industrial SCADA product from the Austrian company ETM (bought by Siemens AG in 2007)



SCADA = **S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition

(commercial software systems used extensively in industry for the supervision and control of industrial processes)

http://www.etm.at/index_e.asp

https://www.winccoa.com/

- Concepts of **WinCC Open Architecture** from www.etm.at (not very helpful):

- Most modern design concepts are used at Simatic WinCC Open Architecture. The continuous object oriented technology sets new standards in SCADA systems. The open concept permits the integration of most different components. Specific solutions from the automation level up to the operating control and management level are provided.

- WinCC Open Architecture features dedicated, autonomous program units for all key functions - the managers. A "manager" is a process that is responsible for specific tasks. For example, there is a separate manager for periphery connections, history data storage or for user interfaces.

- The client-server architecture allows practically unlimited scaling of the system. WinCC Open Architecture is used from small single-site systems to distributed, redundant multi-site systems in a wide range of configurations

Mixed Systems

- At DESY:
  Tango, EPICS, and DOOCS mixed
- At PSI:
  ACS – EPICS migration
- At PSI (former SLS beamline):
  Tango beamline at EPICS accelerator

- There are gateways between the systems

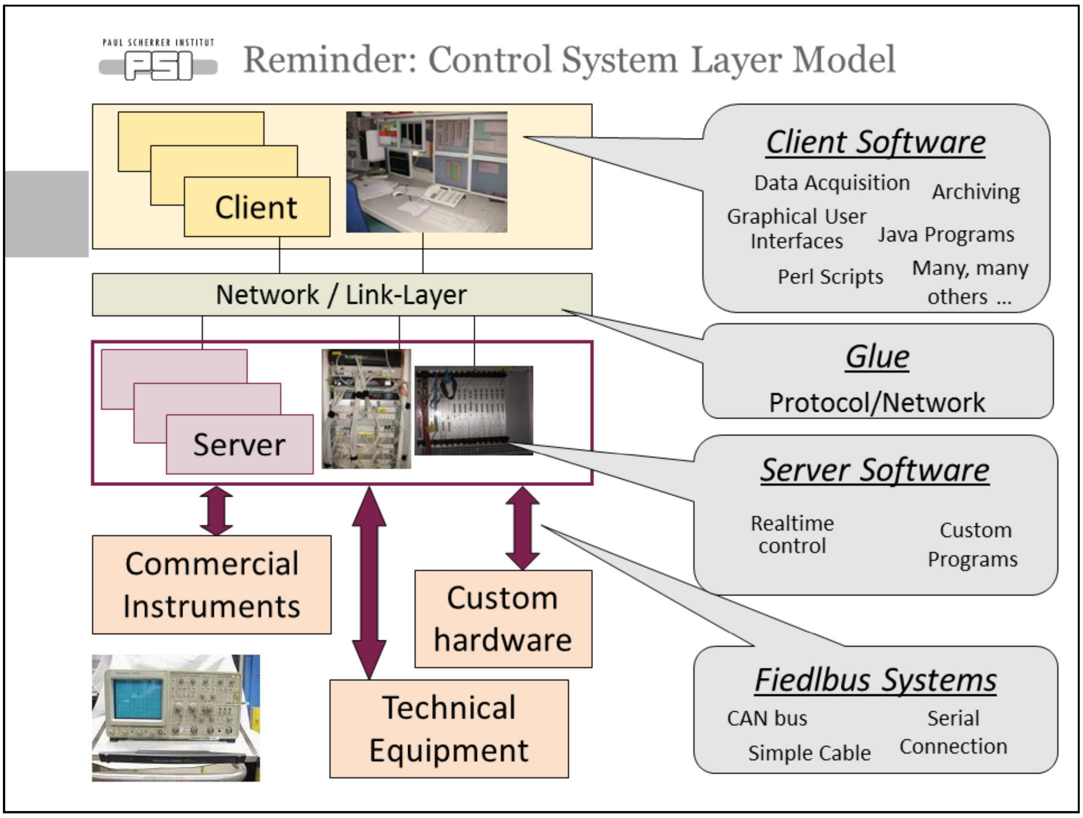
By Evan Swigart

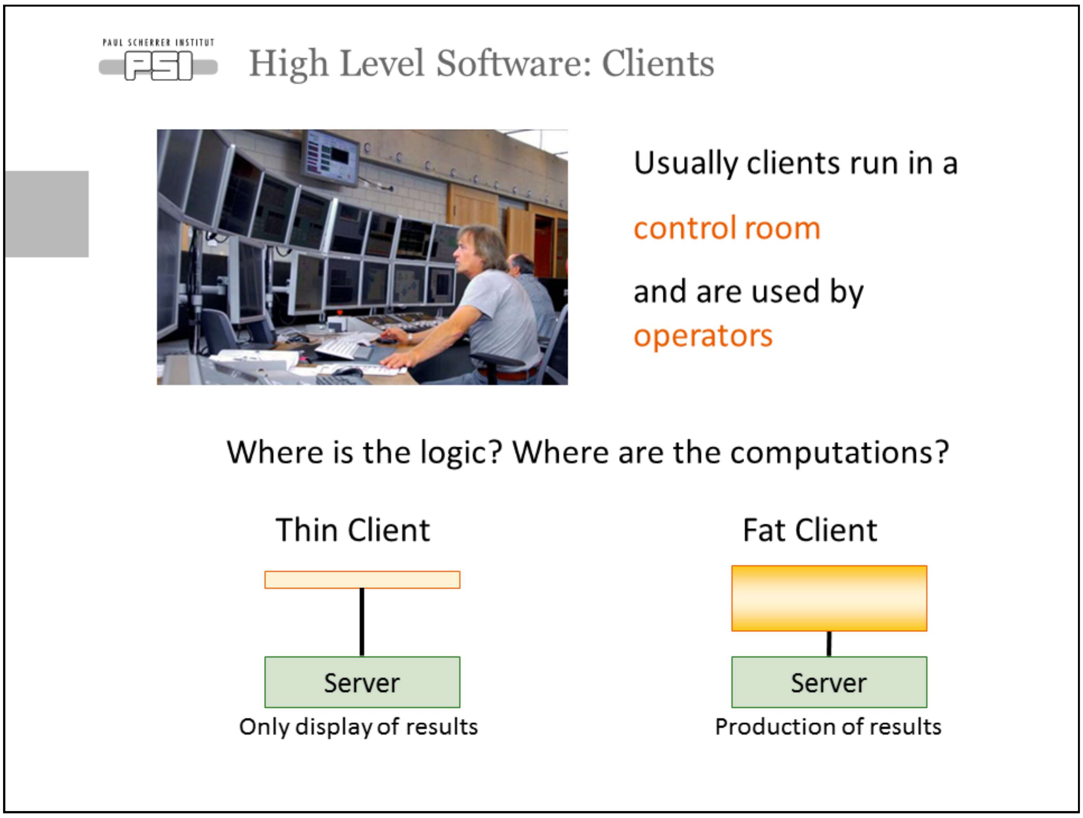
The choice for one system is not exclusive

Table of Content

Technical Requirements

Use open source firmware/software.	• You can change things and you have control of further developments
Use commercial solutions based on open standards developed and sold by a large number of companies	• Don't become dependent on single companies with proprietary solutions
Use standards with a long life-time (20 years+)	• Keep long lifecycles of accelerators in mind


PAUL SCHERRER INSTITUT
PSI
Reminder: Control System Layer Model
Client
Network / Link-Layer
Server
Commercial Instruments
Technical Equipment
Custom hardware
Client Software
Data Acquisition
Archiving
Graphical User Interfaces
Java Programs
Perl Scripts
Many, many others ...
Glue
Protocol/Network
Server Software
Realtime control
Custom Programs
Fiedlbus Systems
CAN bus
Serial Connection
Simple Cable

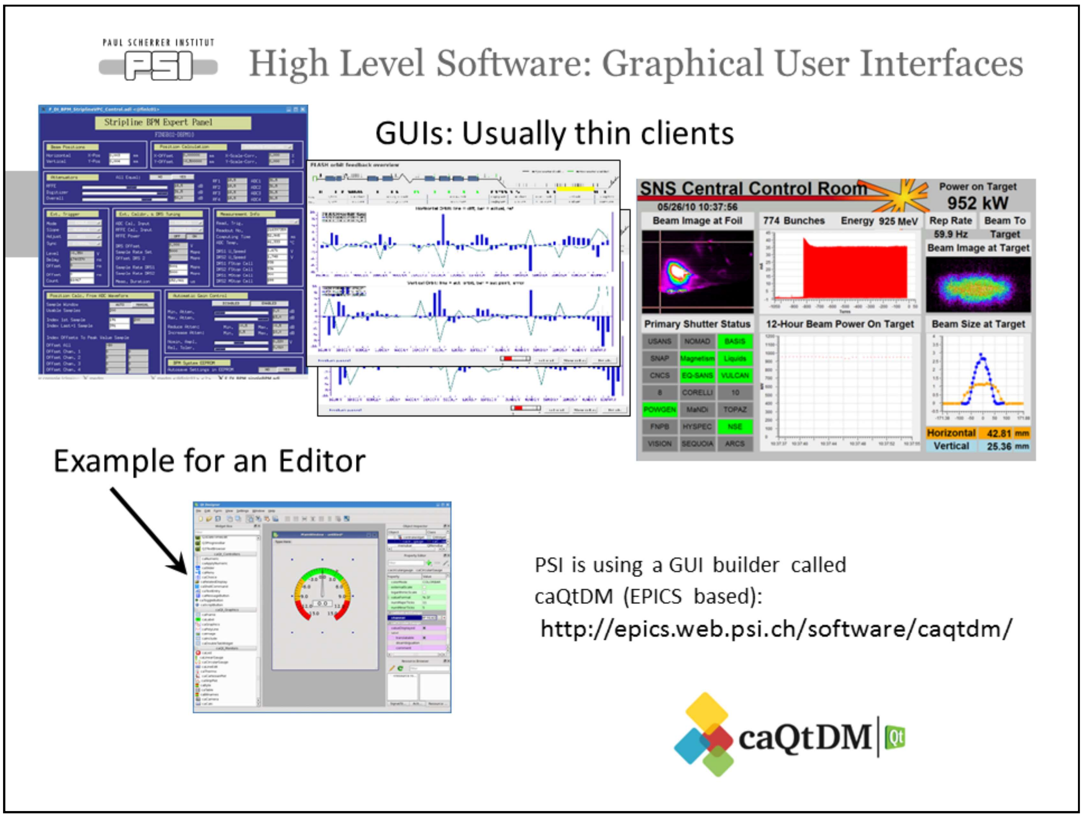


- **Control Room**
- There might be several control rooms: one for the accelerator and one for each beamline or experiment. Or there can be several control rooms around the world in a huge collaboration.

- **Operators**
- There might be dedicated personel for running an accelerator or experiment. Or the scientists and system experts can be on control room duty for some part of their working time.

- **Thin Clients:** Only displays results that are produced in some server elsewhere. Therefore, the display is decoupled from the calculations or logic. Both can be changed independently (for example for updates or new features). Mostly used for expert screens that display the complete interface of the device.

- **Fat Clients:** The results are produced inside the client software. The display is part of the same software. Everything is kept together but no other client can access the results. Usually done by scientists to cover their complete measurement, for example in MATLAB.



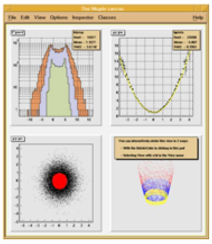
- Home page of JDDD (Java DOOCS Data Display): https://doocs-web.desy.de/index.html#documentation
- BOY (Best OPI, Yet) Operator Interface for Control System Studio (EPICS based): https://github.com/ControlSystemStudio/cs-studio/wiki/BOY

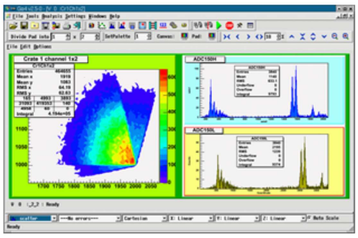
PAUL SCHERRER INSTITUT PSI

High Level Software: Science Applications

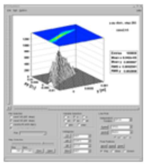
Examples for accelerator science applications:

- Tune measurement and correction
- Orbit correction
- Beam based magnet alignment
- Parameter scans
  (to find optimal working points)
- Filling pattern measurements
  and correction
- Correlation Plots



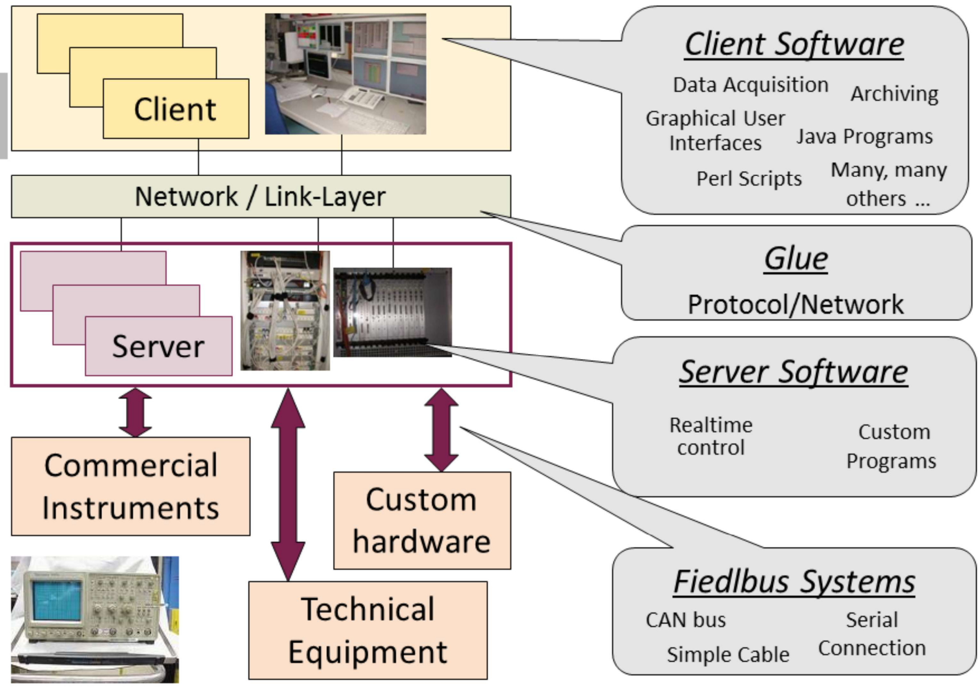


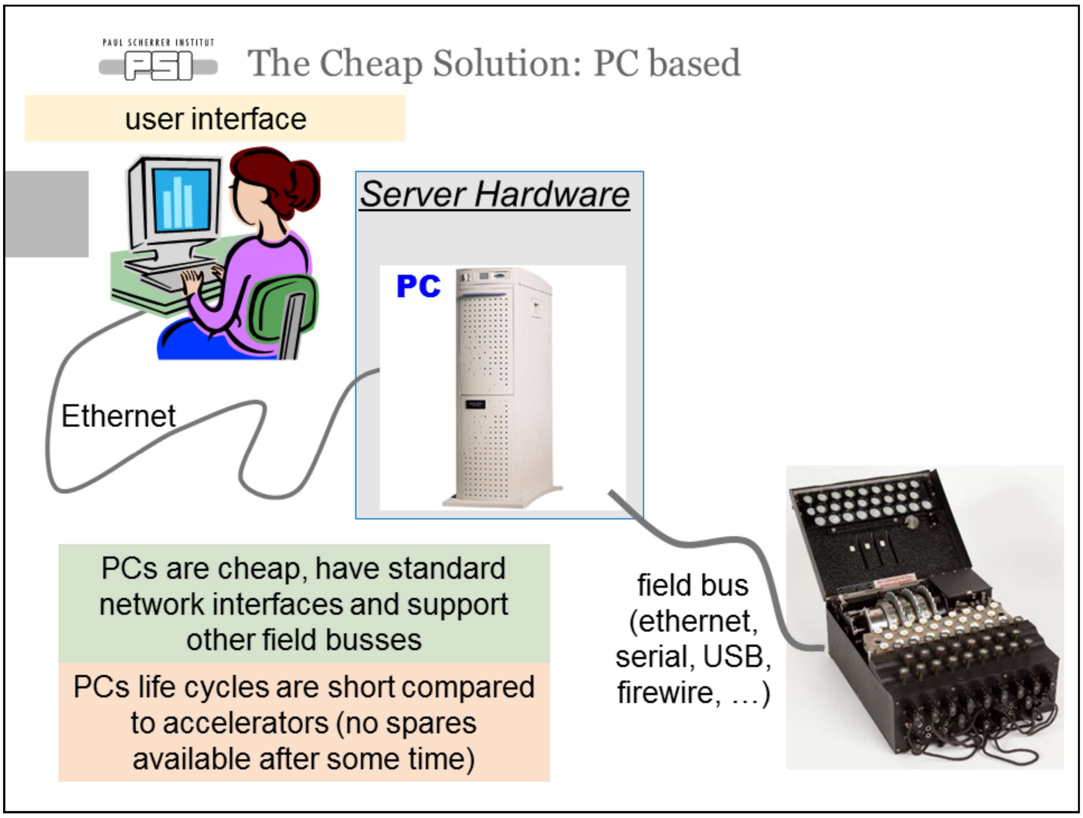
… general data analysis of accelerator data
Usually fat clients,
usually written by scientists (not by controls experts)



- Science Application often use or result in software libraries for specific purposes.
- For example ROOT is an object-oriented program and library developed by CERN (https://root.cern.ch/)
- Other labs use MATLAB from Mathworks for science applications.

Reminder: Control System Layer Model




PAUL SCHERRER INSTITUT
The Cheap Solution: PC based
user interface
Server Hardware
PC
Ethernet
PCs are cheap, have standard network interfaces and support other field busses
PCs life cycles are short compared to accelerators (no spares available after some time)
field bus (ethernet, serial, USB, firewire, …)

PAUL SCHERRER INSTITUT PSI

The Classic Solution: VME based

user interface

Ethernet

Server Hardware

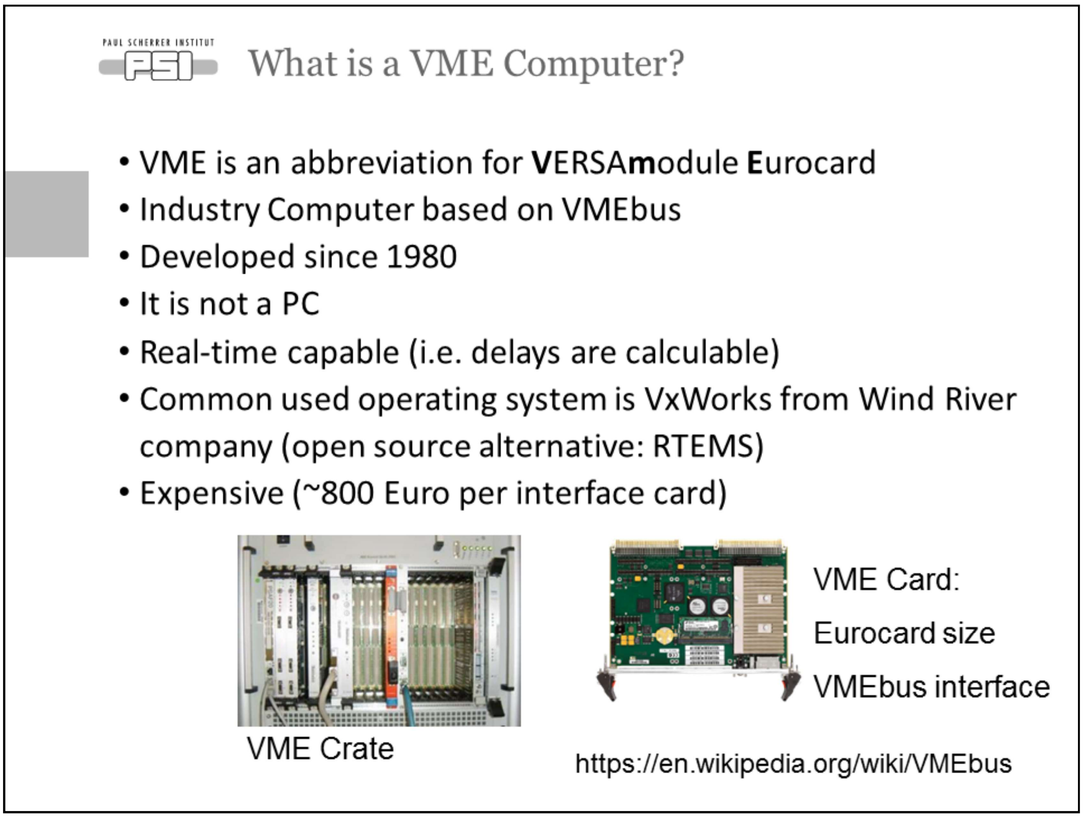
VME
(Operating System: e.g. vxWorks)

Dumb Hardware

Cable or field bus (analog I/O, digital I/O,…)

VME cards life cycle is long,
VMEbus is an open standard,
Supported by Industry

VME is expensive,
special operating system
(VxWorks)



From https://en.wikipedia.org/wiki/VMEbus

Computers using VMEbus include
- HP 743/744 PA-RISC Single-board computer
- Sun-2 through Sun-4
- HP 9000 Industrial Workstations
- Atari TT030 and Atari MEGA STE
- Motorola MVME
- Symbolics
- Advanced Numerical Research and Analysis Group's PACE.
- ETAS ES1000 Rapid Prototyping System
- several Motorola 88000 based Data General AViiON computers
- Early Silicon Graphics MIPS-based systems including Professional Iris, Personal Iris, Power Series, and Onyx systems

A serial interface solution: Picotux based

user interface

Server Hardware

Linux PC

Example for tiny computers with single interface

Ethernet

Serial interface (RS232, …)

Cheap and tiny solution,

Supports distributed devices

All commercial chips have slightly different architecture (maintenance), life cycle yet unknown

The Embedded Solution: Device Integrated CPU

user interface

Embedded Hardware

=

Server Hardware

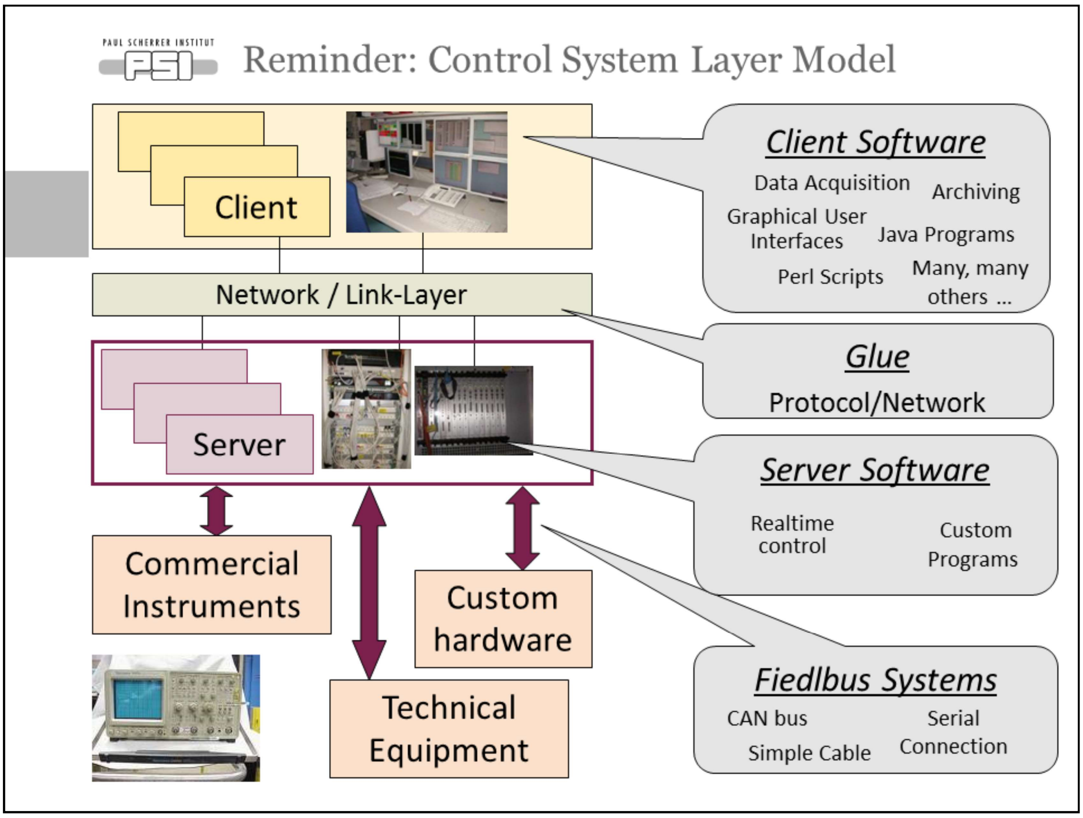
+

Instrument

Ethernet

Low cost, have standard network interfaces and support distributed devices

All commercial chips have slightly different architecture (maintenance), life cycle yet unknown


PAUL SCHERRER INSTITUT
PSI
Reminder: Control System Layer Model
Client
Network / Link-Layer
Server
Commercial Instruments
Technical Equipment
Custom hardware
Client Software
Data Acquisition
Archiving
Graphical User Interfaces
Java Programs
Perl Scripts
Many, many others ...
Glue
Protocol/Network
Server Software
Realtime control
Custom Programs
Fiedlbus Systems
CAN bus
Serial Connection
Simple Cable

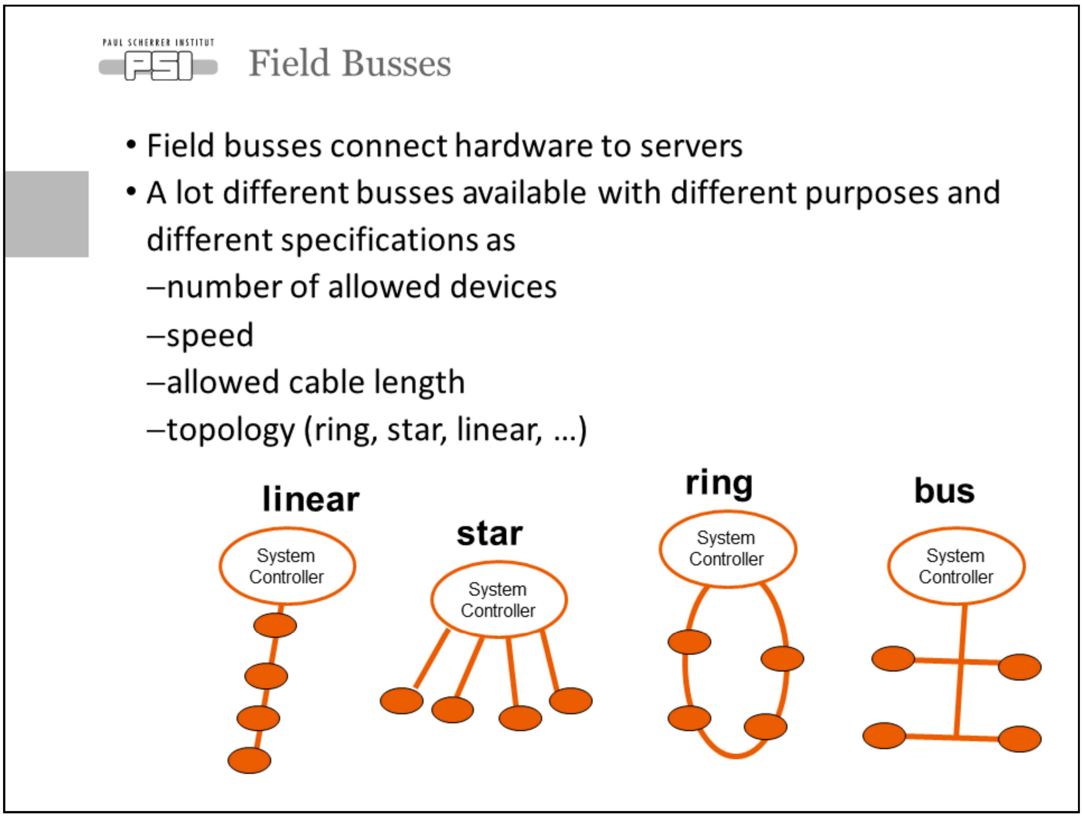
PAUL SCHERRER INSTITUT PSI

What are PLCs?

- **PLC** (**P**rogrammable **L**ogic **C**ontroller)
  – is a digital computer used to connect “dumb” devices

- the PLC is designed
  – for multiple inputs and outputs
  – extended temperature ranges
  – immunity to electrical noise
  – resistance to vibration and impact
  – as a real time system
- Programs are typically stored in battery-backed or non-volatile memory
- Products from different providers can **NOT** be mixed!

- **Examples for companies:**
  - Siemens S7
    (https://www.siemens.com/global/en/home/products/automation/systems/industrial/plc.html)
  - Allen-Bradley
    https://ab.rockwellautomation.com/Programmable-Controllers/
  - Beckhoff
    https://www.beckhoff.ch/



- From Wikipedia:
- **Fieldbus** is the name of a family of industrial computer network protocols used for real-time distributed control, now standardized as **IEC 61158**.

Field Busses

Some example field bus systems:

- **CANbus** (Controller area network)
  https://en.wikipedia.org/wiki/CAN_bus
- **PROFIBUS** (Process Field Bus)
  https://en.wikipedia.org/wiki/Profibus
- **IEEE 1394** (Firewire)
  https://en.wikipedia.org/wiki/IEEE_1394
- **EtherCAT** (Ethernet based real time bus)
  https://www.ethercat.org/en/technology.html

Difference to Ethernet and USB?
Field busses are real time capable (IEC 61158 specification)

- **Definition of Real-time:**
- The worst case for the reaction time from signal in to reaction out can be determined. Real time is not necessarily fast (but often).

- **Ethernet** and **USB** are used in place of field busses in accelerators where no real time capability is needed.

PAUL SCHERRER INSTITUT PSI

Table of Content

PAUL SCHERRER INSTITUT PSI

Borderlands of Control Systems

Accelerator Control Systems have fussy borders.

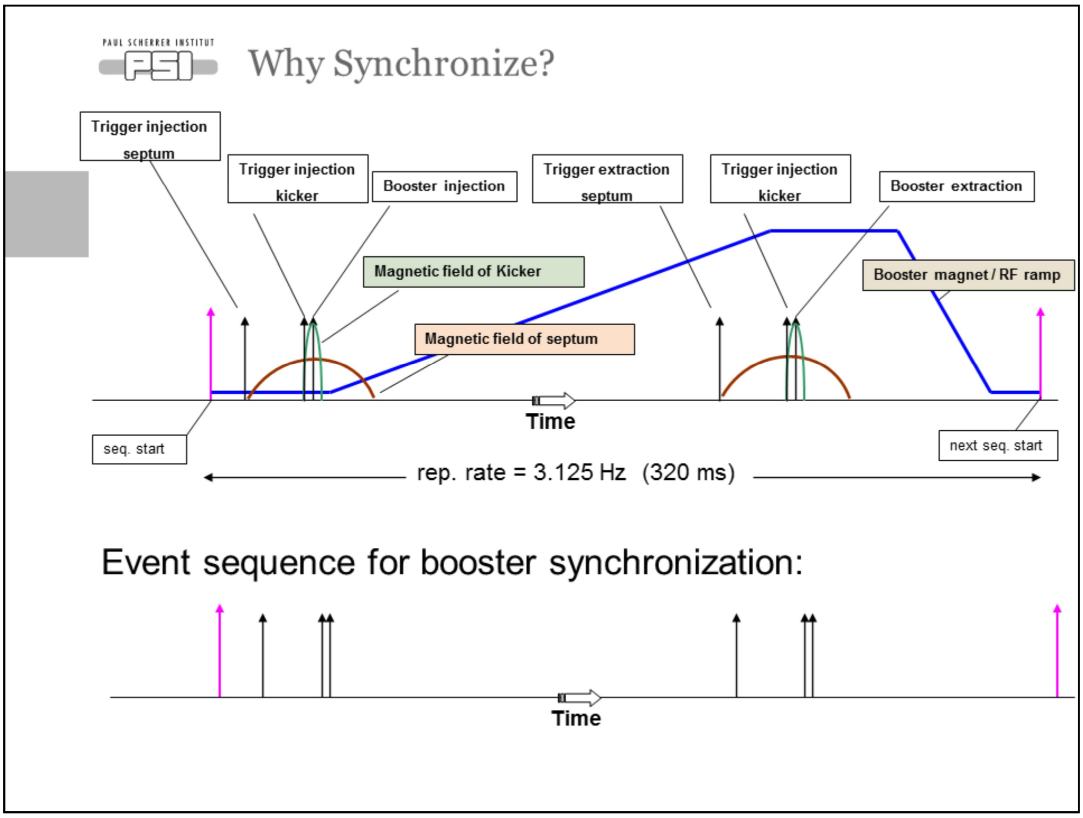
Some example for these borders are:

1. Timing and Synchronisation
2. Feedback Systems
3. Interlock-, Alarm-, and Machine Protection Systems
4. Experiment Data Acquisition
5. Relational Databases
6. Relationship of IT (Information Technology) and Controls

- All of these fields are needed in accelerator control. For all of these topics the responsibilities have to be clarified.

For example

1. Timing and Synchronisation



- Triggers might be needed in some advanced time to allow real hardware to ramp up or get the full field.
- Especially important timing and synchronisation are for pulsed accelerators like FELs.

- The example is taken from the booster synchrotron timing from the Swiss Light Source (PSI).

Solutions for Timing Systems

- Master oscillator + delay cables
  (1 trigger and measured cable lengths)

- Master oscillator + digital delay generators
  (https://www.thinksrs.com/products/DG535.htm)

- (Master oscillator +) event generators/receiver cards in computers (PC , VME, µTCA)
  (http://www.mrf.fi/)

- Timing and synchronization is needed to run an accelerator
- Various solutions available and used

Timing and synchronization can be part of the Control System.

Clarify who is responsible for timing and synchronization to avoid problems!

- Accuracy of the timing signal can vary from some pico seconds down to femto seconds – but femto seconds are really hard to get: http://accelconf.web.cern.ch/accelconf/ipac2012/papers/weeppb006.pdf

- Technology depends on needed accuracy and available budget.

- Other solutions exist: https://en.wikipedia.org/wiki/White_Rabbit_Project

White Rabbit is the name of a collaborative project including CERN, GSI Helmholtz Centre for Heavy Ion Research and other partners from universities and industry to develop a fully deterministic Ethernet-based network for general purpose data transfer and sub-nanosecond accuracy time transfer. Its initial use was as a timing distribution network for control and data acquisition timing of the accelerator sites at CERN as well as in GSI's Facility for Antiproton and Ion Research (FAIR) project.

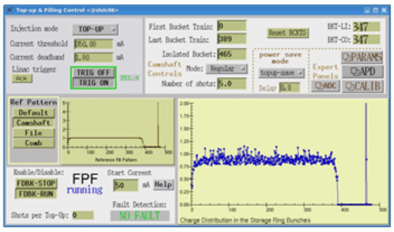
For example

2. Feedback Systems

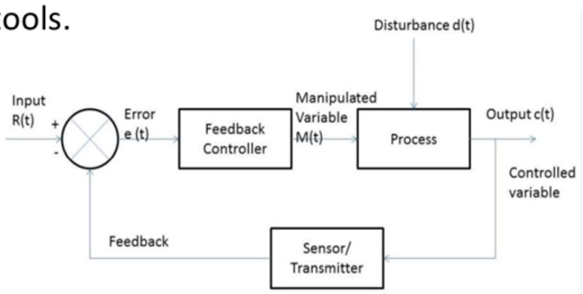
PAUL SCHERRER INSTITUT

Feedback Systems

For example:
- Orbit Feedback (Position)
- Energy Feedback
- Filling pattern Feedback



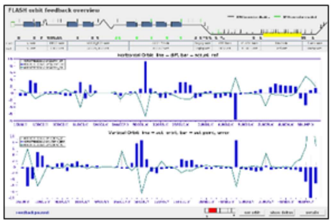
If it needs to be fast, it needs separate cables!
Slow feedbacks can be realised with standard control system tools.



- See also https://en.wikipedia.org/wiki/PID_controller
- A proportional–integral–derivative controller (PID controller or three-term controller) is a control loop mechanism employing feedback that is widely used in industrial control systems and a variety of other applications requiring continuously modulated control. A PID controller continuously calculates an error value e(t) as the difference between a desired setpoint (SP) and a measured process variable (PV) and applies a correction based on proportional, integral, and derivative terms (denoted P, I, and D respectively), hence the name.

Example: Orbit Feedback

Needed for beam position stability.



Measurement (once in a time):

- Measure beam response matrix (complete orbit for different corrector magnet settings)
- Invert the matrix (normally not possible analytical, use numerical methods) a stable method is singular value decomposition (SVD)

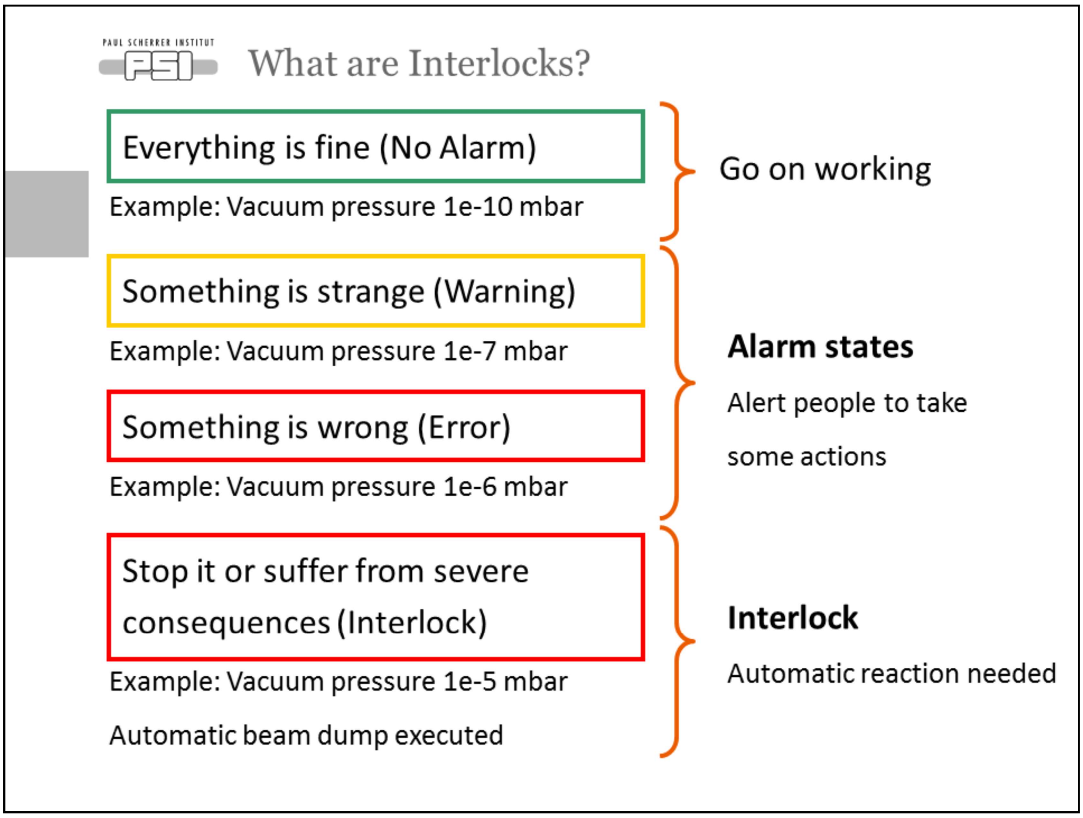
Feedback during runs:

- Measure the beam position and correct it with the appropriate set of correctors

- See for example https://en.wikipedia.org/wiki/Singular-value_decomposition

For example

3. Interlock-, Alarm-, and Machine Protection Systems



- All interlock and safety systems need to be hard wired and not involve software (if possible).
- Training of the operators need to include alarm handling. As often several alarms occure at the same time (or at the same time as far as a human can see) this can be complicated. For running machines, alarm handling is one of the two main tasks of the operators (the other is setting up and tuning of different machine modes).
- The example vacuum pressures are taken from the Swiss Light Source.

PAUL SCHERRER INSTITUT PSI

What are Alarms

EPICS Alarmhandler

Murphy's law:
Anything that can go wrong
will go wrong.

- Alarms help to avoid Real Problems
- Alarms help to find problems
- Example:
  - –Beam position more than 1 mm of from reference
  - –Vacuum pressure higher than 1e-6 mbar
  - –Orbit Feedback Program not running

- People should react on alarms

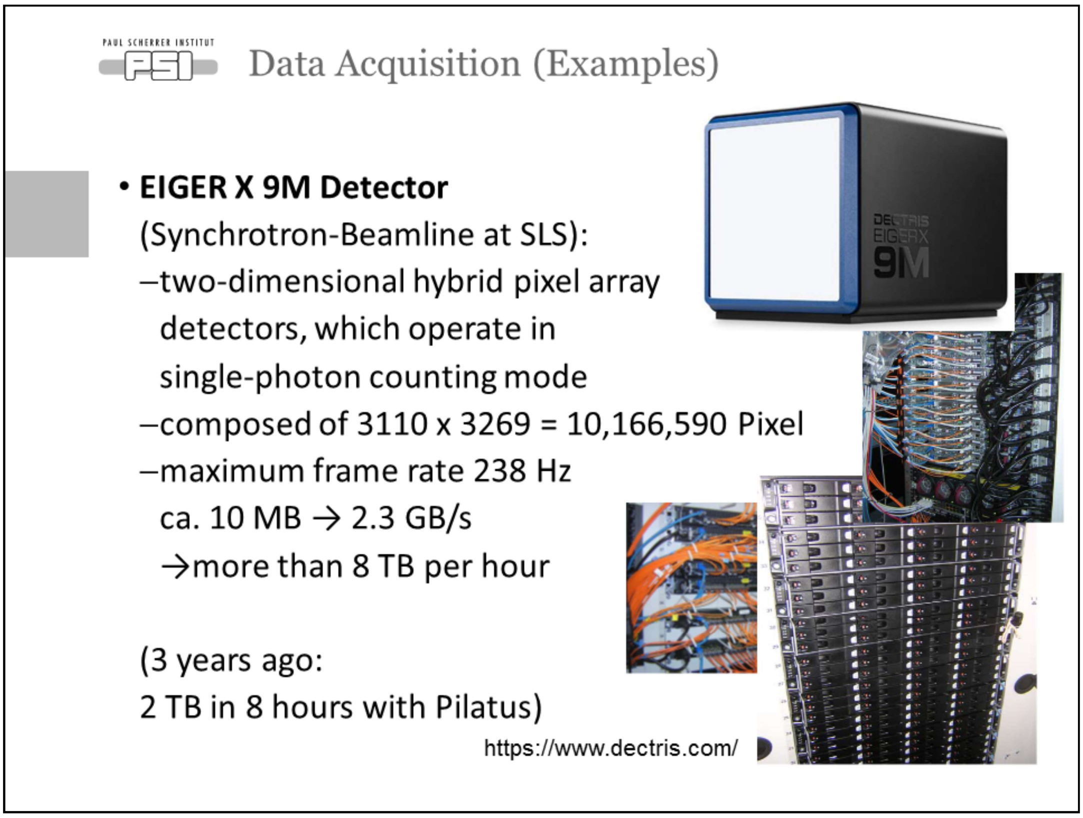
Interlock Systems

- Interlock Systems have to be
  - taking automatic actions (no people involved) - fast
  - Reliable (99% might not be enough)
  - as simple as possible (see Murphy's law)
- Avoid computers in Interlock Systems
  Decouple "running" the accelerator (=Control System)
  from "stopping" the accelerator (=Interlock System)
- There can/will be more than one Interlock System in an accelerator (local, global, different goals, etc.), for example:
  - Vacuum Interlock
  - Equipment Protection System
  - local RF Interlock Systems

Clarify who is responsible for Interlock Systems to avoid problems!

For example

4. Experiment Data Acquisition



- For basic funcionality see https://en.wikipedia.org/wiki/Hybrid_pixel_detector

Data Acquisition (Examples)

- The Large Hadron Collider will produce roughly 15 petabytes (15 million gigabytes) of data annually – enough to fill more than 1.7 million dual-layer DVDs a year!
  - GRID computing to allow access

https://en.wikipedia.org/wiki/Worldwide_LHC_Computing_Grid

- See also https://wlcg.web.cern.ch/ for Worldwide LHC Computing Grid

Is Data Acquisition Controls?

- Data Acquisition requires
  - Network infrastructure
  - Computer storage infrastructure
  - Server infrastructure for data access
  - Environment (e.g. Grid) for data access
  - Manpower for setup and maintenance
- Detectors
  - can provide information about accelerator (beam position)
  - need to be adjusted to accelerator setup (connection to control system needed)
- Some detectors (e.g. BPMs) are part of the accelerator anyway

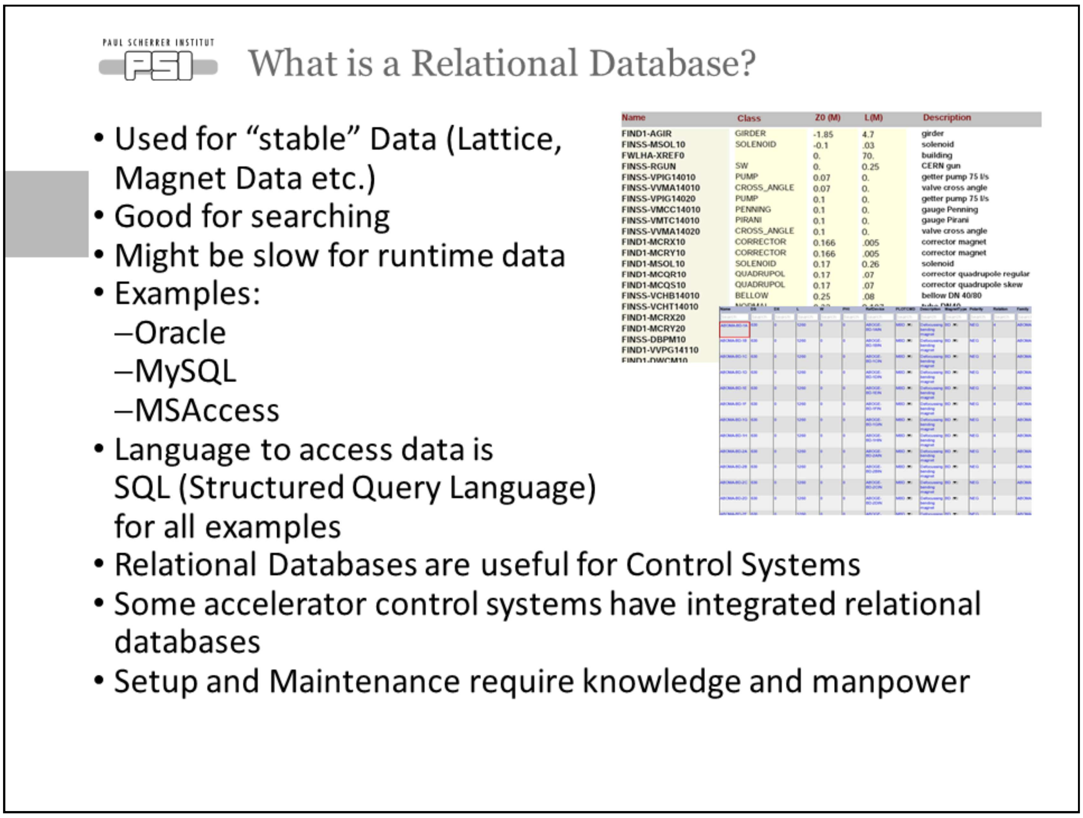
Not necessary

Yes its needed

Has to be discussed to avoid problems!

For example

5. Relational Databases



• In most cases the setup of the database framework (i.g. installation of the database) is consideret an IT task. The creation, composition, and fill up with data might be considered a controls task (or even a task for beam dynamics physicists).

For example

6. Relationship of IT (Information Technology) and Controls

Who is Responsible for What?

- Most large research institutes have a Controls Group in addition to a IT Group
- Why separate IT from Controls?

IT	Controls
• Office PC installation	• Accelerator computer installation
• Operating Systems for Office applications	• Integration of accelerator hardware
• Infrastructure (network cables)	• Control Room applications
• Central Services (Computing Cluster, Server Room …)	• Distributed processes

Databases, Timeserver, Network, Security

Controls is dependent on IT.

Responsibilities have to be discussed to avoid problems!

- IT = Information Technology (includes Computers but also Telephone, Mobile …)

Table of Content

Summary: What is Accelerator Controls

- It is hard to define – but every Accelerator has one
- It is organized in layers separating hardware from applications
- It is (has to be) a distributed system, involving some network protocols
- The borders are not clearly defined
  - For example: Where starts the hardware responsibility (PLCs, embedded systems)?

Definition:

An **Accelerator Control System** is a computer environment that allows remote access to the accelerator hardware with a lot of different functionality to satisfy the requirements of several different user groups.

- It is not considered science but needs a lot knowledge about science (physics and computer science, sometimes politics)

References

Bad news: There is no book on Accelerator Control Systems
Good news: You can find some things in the Internet

- ICFA Newsletter Number 47 (December 2008) on Control System: http://icfa-usa.jlab.org/archive/newsletter/icfa_bd_nl_47.pdf

- EPICS: https://epics-controls.org/
- Tango: http://www.tango-controls.org/
- CERN Controls Group: https://be-dep-co.web.cern.ch/
- PSI Controls Group: http://epics.web.psi.ch/

…search the institute web pages …

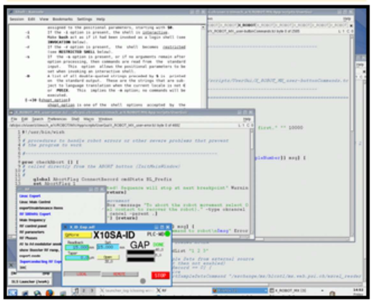
- International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS): https://www.icalepcs.org/

What to Learn as a Controls Guy?

1. Be curious about what your customers do (accelerator physics, experiments, medical treatment, etc.)
2. Enjoy programming
   - Script Language (python or similar)
   - Object Oriented (Java, C++, etc.)
3. Enjoy computer environments

Quick test:

Do you feel comfortable with this



- Useful skills include (non-essential)
  - Basic knowledge in Accelerator Physics or general Physics
  - Database structures/sql commands
  - Linux and/or Windows administration
  - Network administration
  - PLC, FPGA or DSP programming (nearly electronics)
  - Graphical User Interface design

PAUL SCHERRER INSTITUT PSI

Wir schaffen Wissen – heute für morgen