Mastering EPICS

Essential Technologies and Architectures

MAURO GIACCHINI

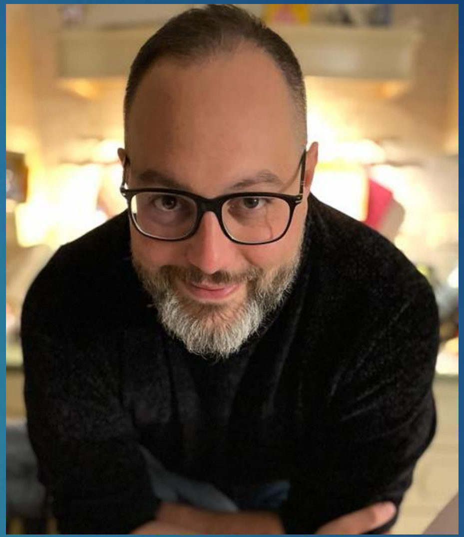
MAURIZIO MONTIS

EPICS SUMMER SCHOOL

BERLIN, 05TH AUGUST 2024

Who We are

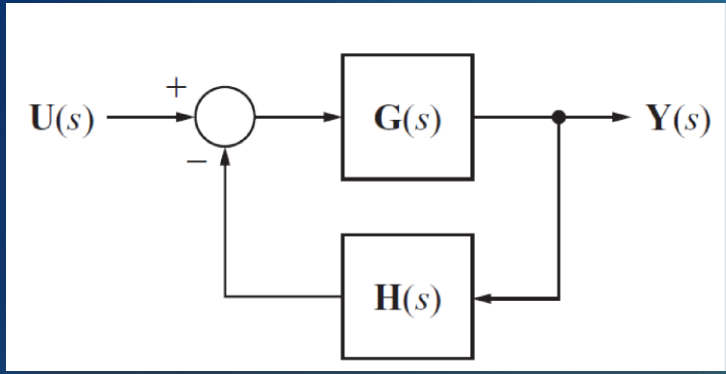
- **Mauro Giacchini**
- My first EPICS Meeting Epics Meeting in 1998 took place at BESSY II
- ...and somehow still sane!
- Grateful to be presenting where my EPICS journey started.



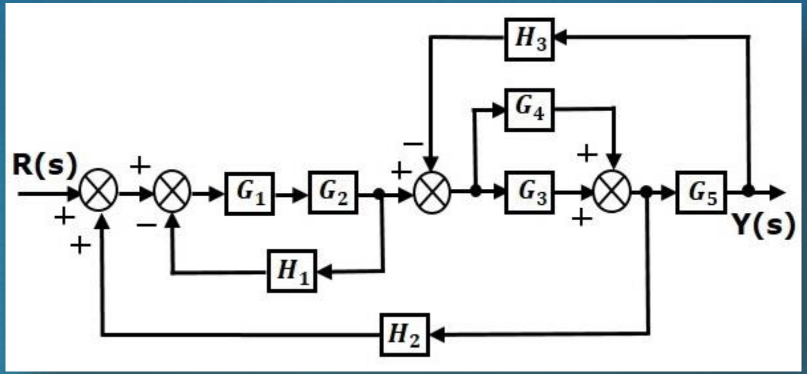
- **Maurizio Montis**
- Control System Engineer from UniPD (Italy)
- Involved in EPICS since 2011
- …2009 if I consider the MS Thesis (but I was young)
- Motto:

What is a Control System?

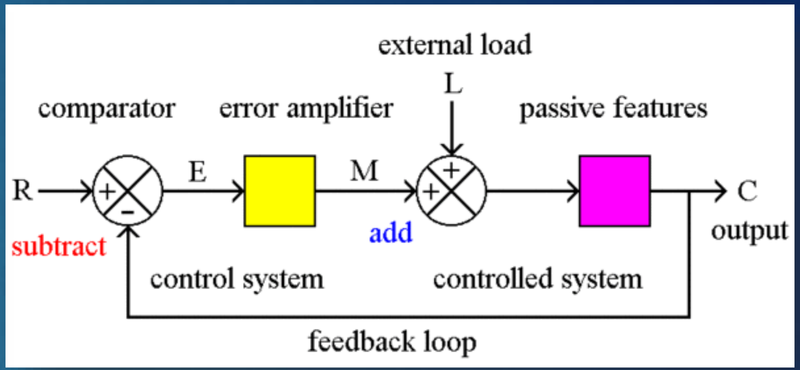
Control Systems - Theory



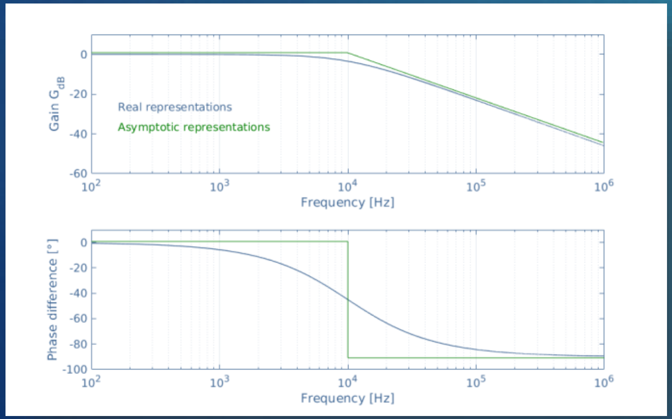
Mathematical modelling



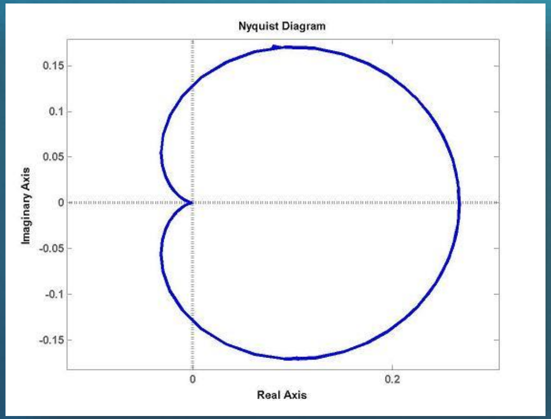
More complex mathematical modelling



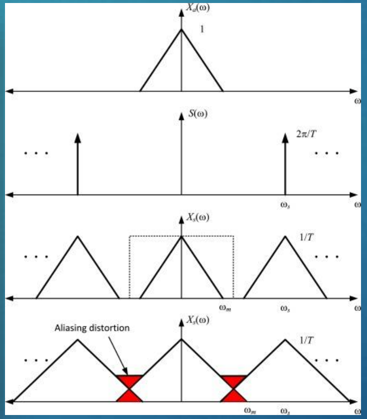
Feedback loop design



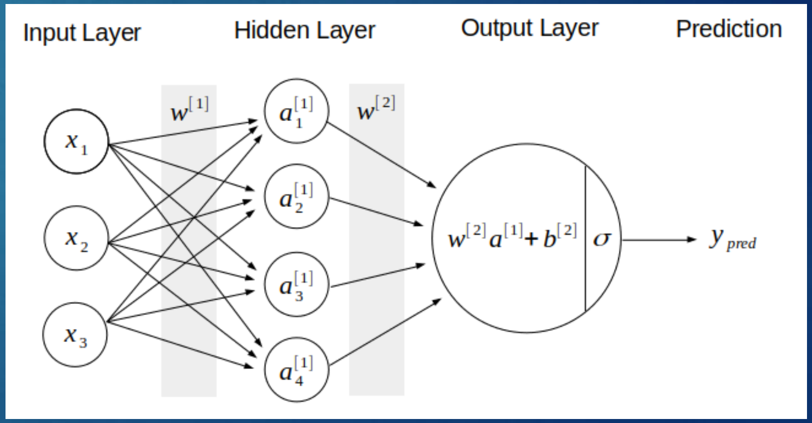
Bode Diagrams



Nyquist Diagrams



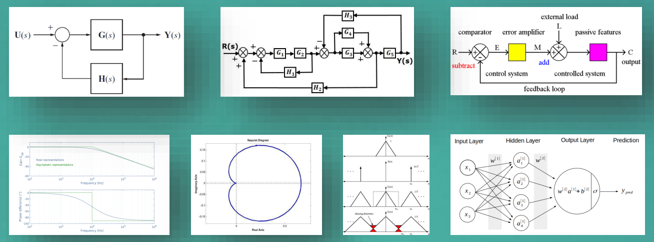
Shannon Sampling Theorem

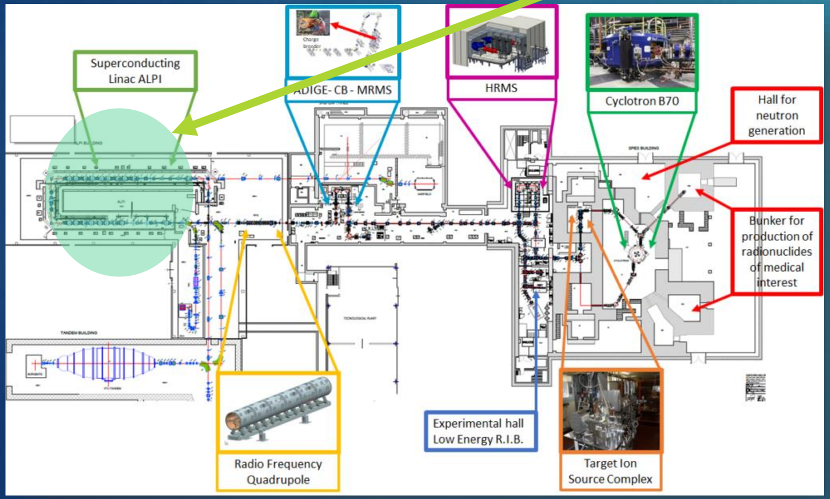


AI / ML / Neural Networks and beyond

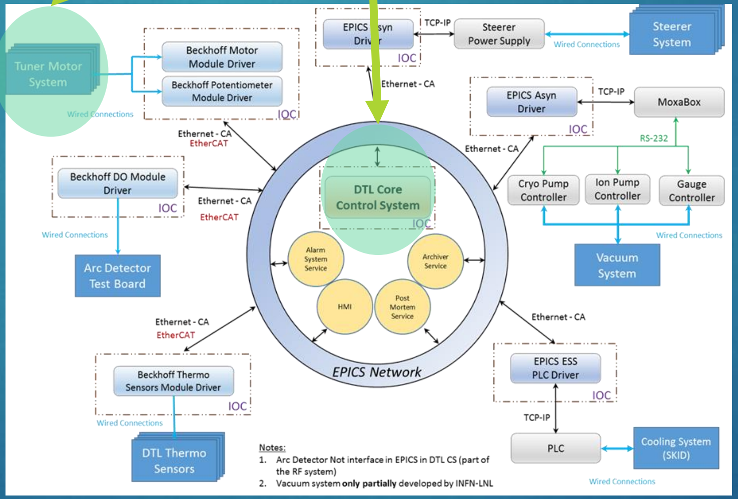
Control Systems – not only Theory

Theory

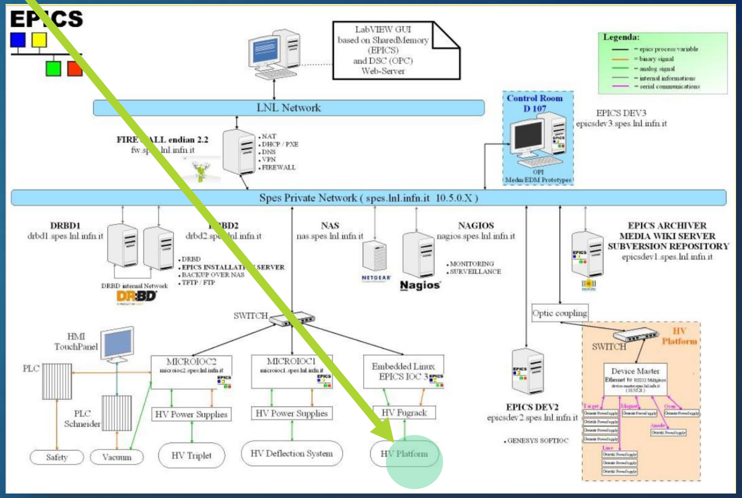




(Very) Complex System Modelling

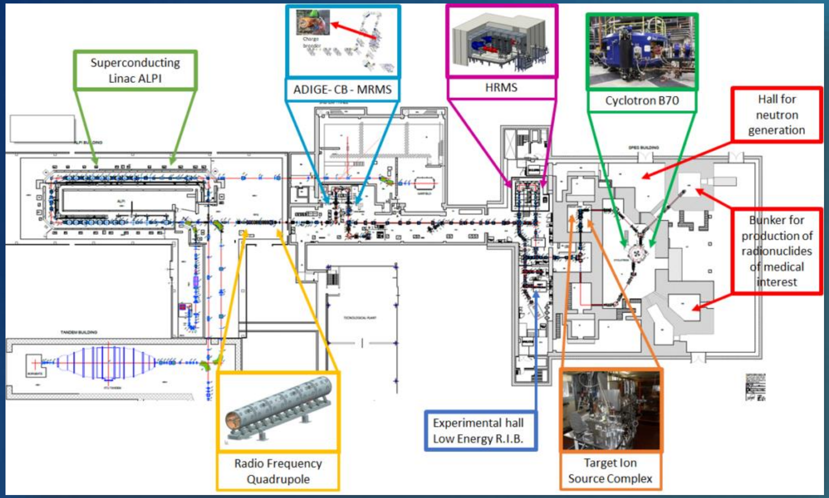


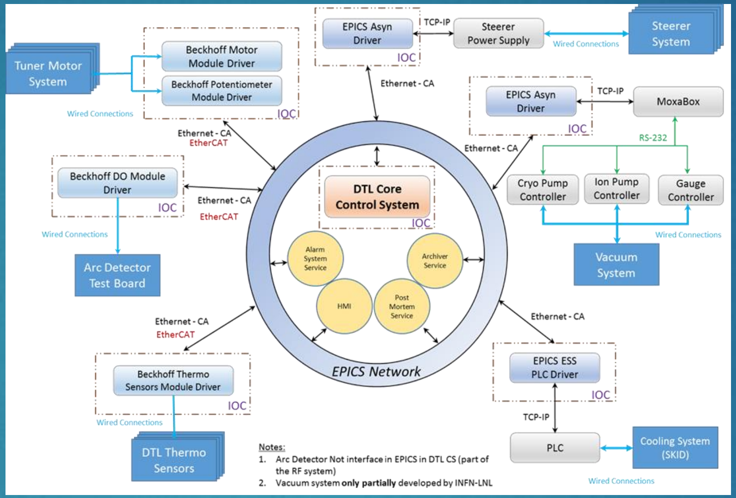
Heterogeneous kind of Hardware, Software, Protocols

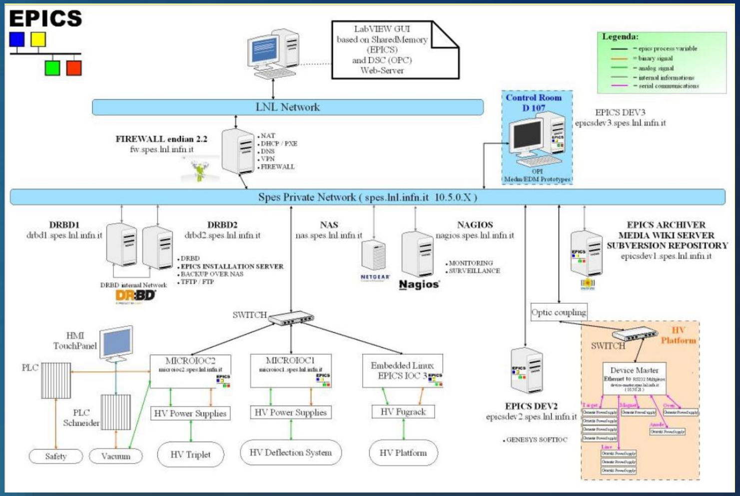


Control System Architectures

Control Systems Scenarios







(Very) Complex System Modelling

Divide-and-conquer approach

- R&D of a single apparatus / functional system
- High-level orchestraton

Heterogeneous kind of Hardware, Software, Protocols

Every single apparatus, and the entire facility, require and use different hardware and software solutions

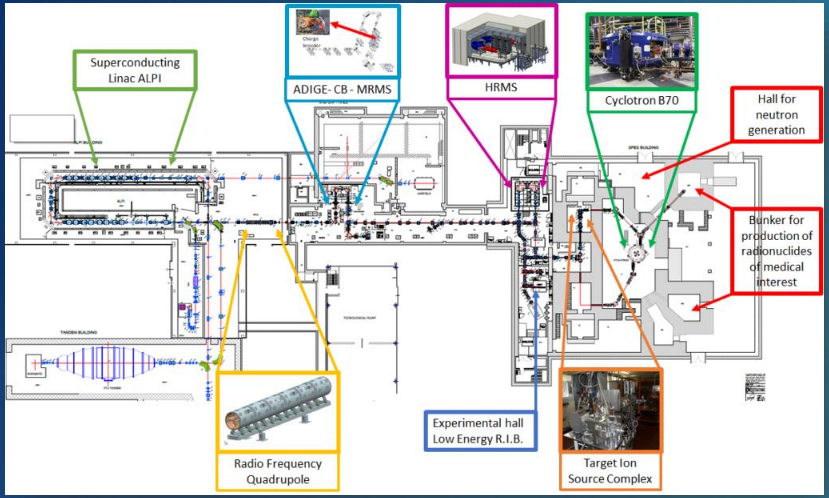
Many times, hardware limits the options at software level

Control System Architectures

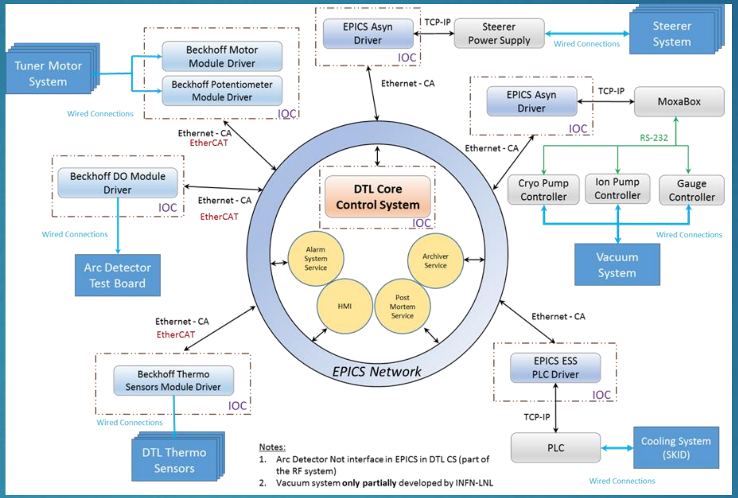
A control system engineer must have multiple knowledges in:

- Networking and IT architecture
- Protocols and field buses
- HW interfaces
- High-level services and GUIs design

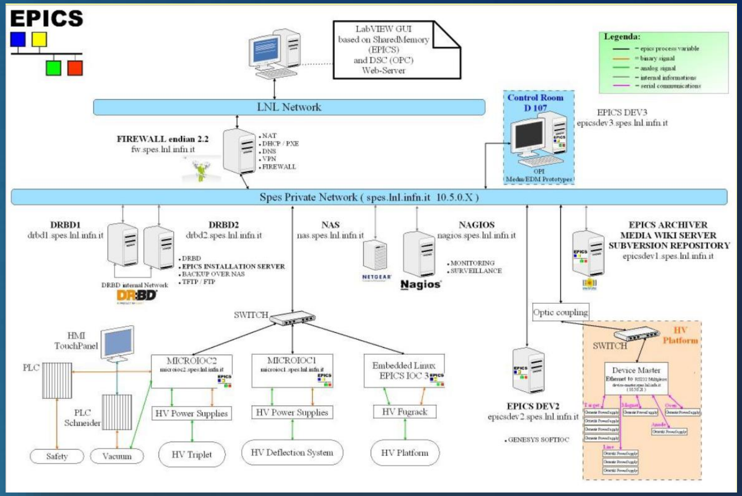
Control Systems



Many different apparatus

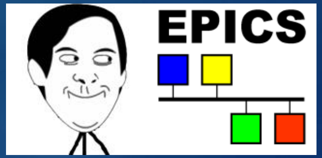


Heterogeneous kind of Hardware, Software, Protocols



interface to the End User(Scientist, Operator, etc.) though proper services and applications

EPICS



Contents

- Thomas has already done an excellent intro about CS
- EPICS in a Nutshell
- Networking
- Hardware Interfaces and Devices
- Field Buses
- IT architecture
- Software Tools and Middleware
- Human-Machine Interface (HMI)
- Documentation and Compliance

What is EPICS?

- **A collaboration** *(Day 2, Ralph's talk)*
  - A worldwide collaboration that shares designs, software tools, and expertise for implementing large-scale control systems
- **A control system architecture**
  - A client/server model with an efficient communication protocol (Channel Access) for passing data
  - A distributed database of machine values
- **A software toolkit**
  - A collection of software tools collaboratively developed which can be integrated to provide a comprehensive and scalable control system

Collaboration

Not exhaustive list of contributors:

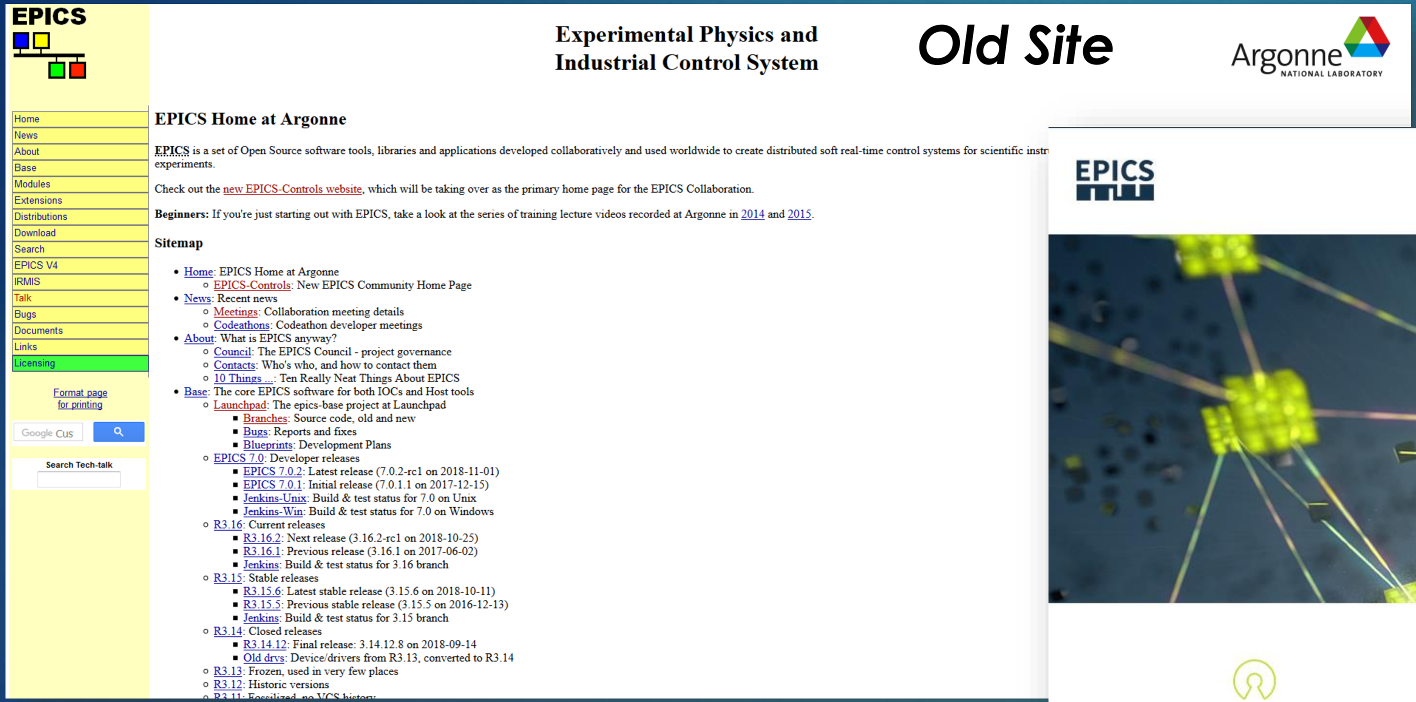
- ANL (USA)
- LANL (USA)
- ORNL (SNS) (USA)
- SLAC (USA)
- SSRL
- LCLS
- JLAB (USA)
- CEBAF
- DESY (Germany)
- HZB (ex BESSY) (Germany)
- PSI (Switzerland)
- SLS
- SwissFEL
- Gemini (USA, Chile)
- Keck (USA)
- KEK (Japan)
- Diamond (UK)
- AS (Australia)
- NSSRC (China)
- BNL (USA)
- INFN (Italy)
- ASKAP (Australia)
- ITER (France)
- ESS (Sweden)
- IFMIF (Japan)

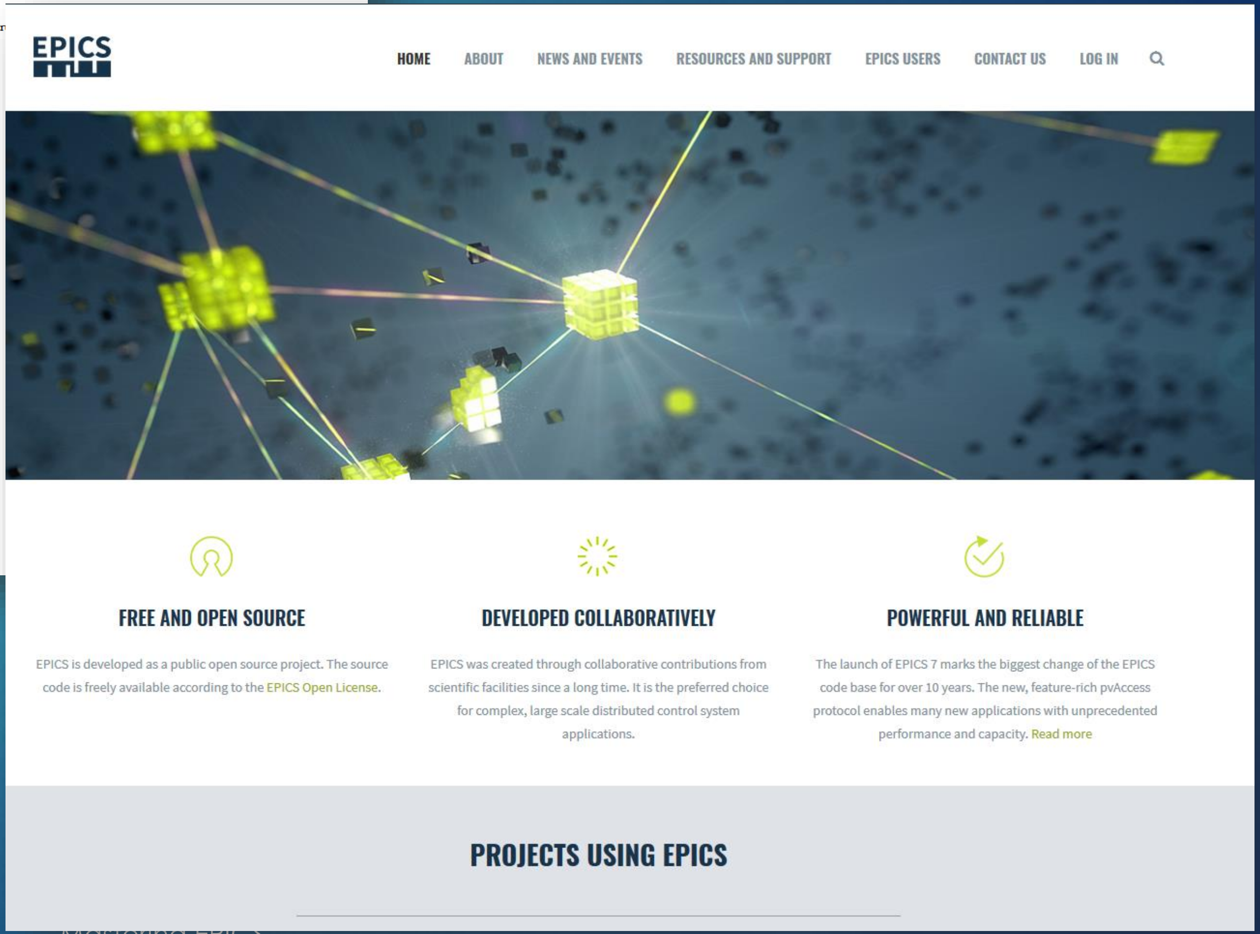
Collaboration

- Began in 1989 between LANL/GTA & ANL/APS (Bob Dalesio & Marty Kraimer)
- Over 150 license agreements were signed before EPICS became “open source” in February 2004
- Regular collaboration meetings
  - 2 meetings per year
  - 100-200 participants
  - Over 70 presentations (3+ days) for single meetings
- Home page with links to manuals, presentations, software
  - https://epics.anl.gov/ and https://epics-controls.org/
- List server; tech-talk: the collaboration in action
- Collaborative efforts vary
  - Assist in finding bugs
  - Share tools, schemes, and advice

EPICS sites

Starting point for documentation





- Old Site: https://epics.anl.gov/
- New Site: https://epics-controls.org/

Speaking in… EPICS

- **EPICS**

  Experimental Physics and Industrial Control System
- **CHANNEL ACCESS (CA) / PVACCESS (PVA)**

  Communication protocol used by EPICS / New Communication protocol used by EPICS 7
- **PROCESS VARIABLE (PV)**
  1. A piece of named data referred to by its PV name
  2. The primary object of the channel access protocol
- **CHANNEL**

  A synonym for Process Variable
- **CHANNEL ACCESS SERVER**

  Software that provides access to a PV using the CA protocol
- **CHANNEL ACCESS CLIENT**

  Software that requests access to a PV using the CA protocol

Speaking in... EPICS

- **INPUT OUTPUT CONTROLLER (IOC)**

  A computer running a set of EPICS routines used to define PVs and implement (real-time) control algorithms

  IOC uses database records to define PVs and their behavior

- **SOFT IOC**

  An instance of IOC running as a process on a "non-dedicated" computer, such as a general purpose computer

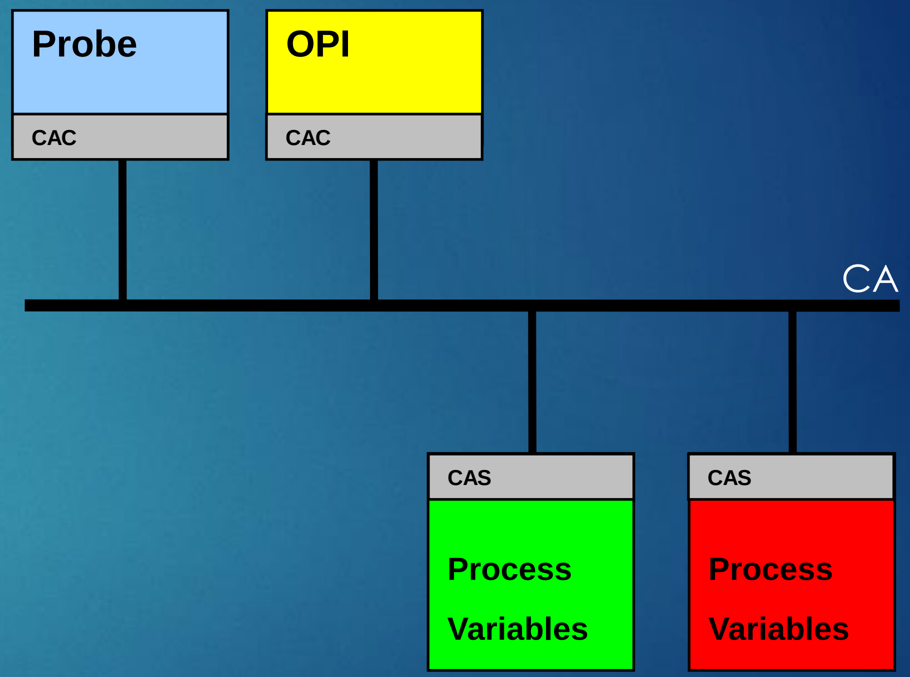
- **RECORD**

  The mechanism by which a PV is defined in an IOC

  Many record types available, each with it's own attributes and processing routine that describe its functionality

Control System Architecture

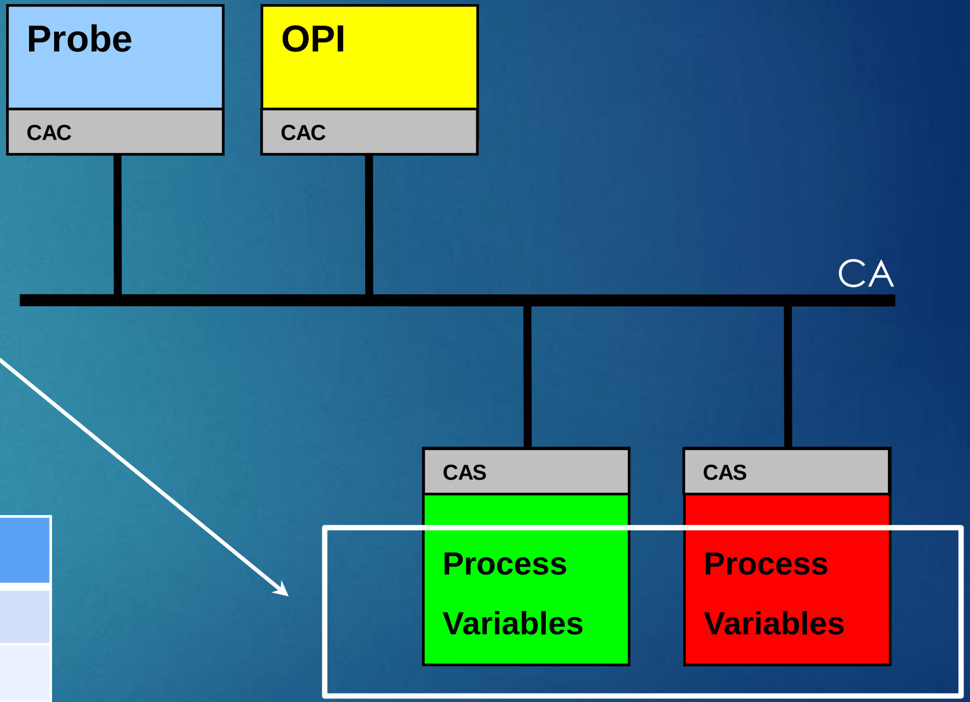
- Network-based "client/server" model where EPICS servers (CAS) and clients (CAC) "speak" Channel Access (CA)
- EPICS is a distributed control system **where every client can see all the servers**
- The "service" that a Channel Access server provides is access to a Process Variable
- The Process Variable (PV) is a named piece of data.
  The name must be unique!



Control System Architecture

- The **PV** is a named piece of data associated with the machine
  - status
  - readback
  - setpoint
  - parameter
- Examples of PV names and values:

PV NAME	*VALUE*
RFQ:VAC:Pressure	6.2e-08 mBar
LINAC_BPM4:xPosition	-0.117 mm
BOOSTER:gateValvePosition	'OPEN'
Line3:DIPOLE_PS-setPoint	502.7 Amps
LINE_Ctlr:Mode	'Warm Up'
BOX2:HIST	{13, 8, 21, 22, 56, 44, 32, 43, 33, 15, 11}

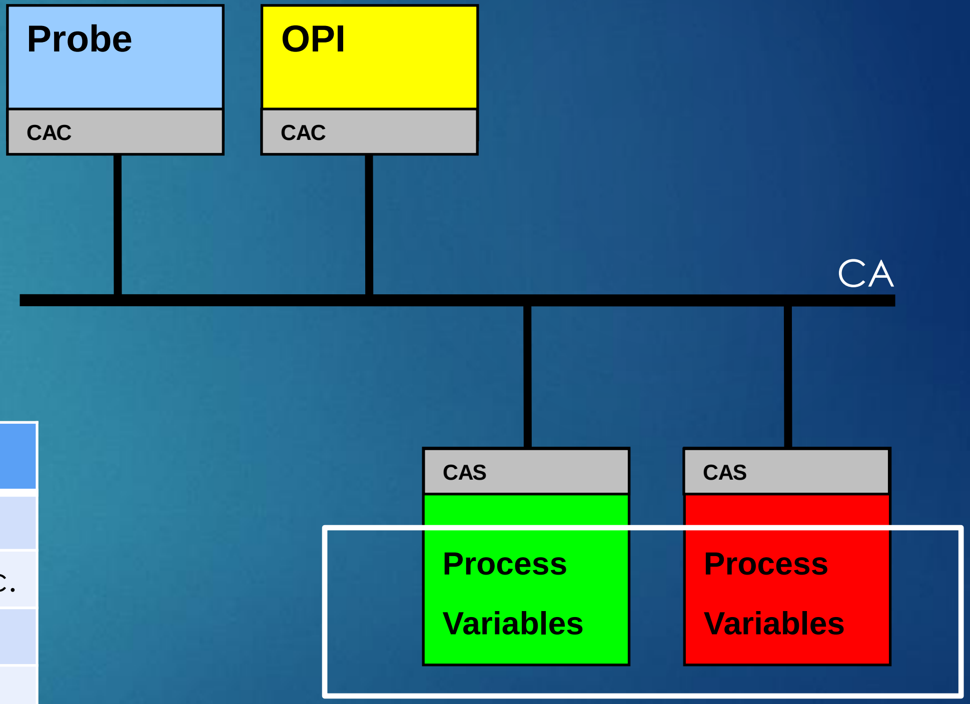


Control System Architecture

- Every **PV** has a set of attributes
  - Common attributes
  - Type-based attributes

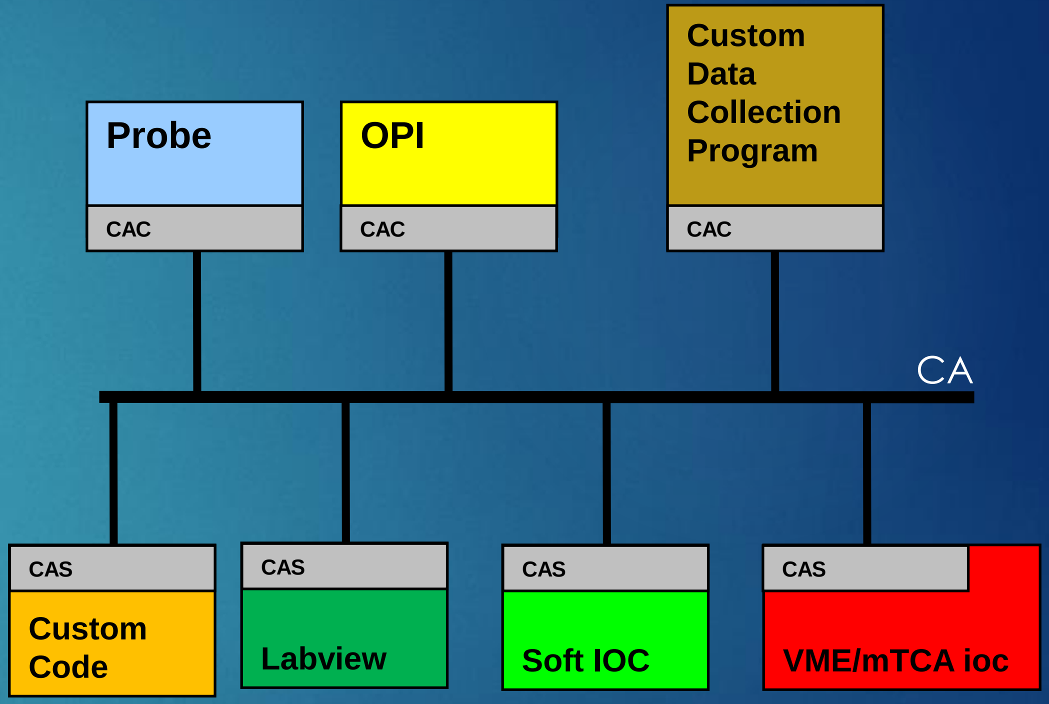
- Examples of PV names and values:

ATTRIBUTES	*Note*
Alarm Severity	NO_ALARM, MINOR, MAJOR, INVALID
Alarm Status	LOLO, LOW, HIGH, HIHI, READ_error …etc.
Timestamp	
Number of Elements (array)	
Calculation Condition	A>B?A+B:C
Control Limits	
Engineering Unit	"Volt", "A", "%", "mm"



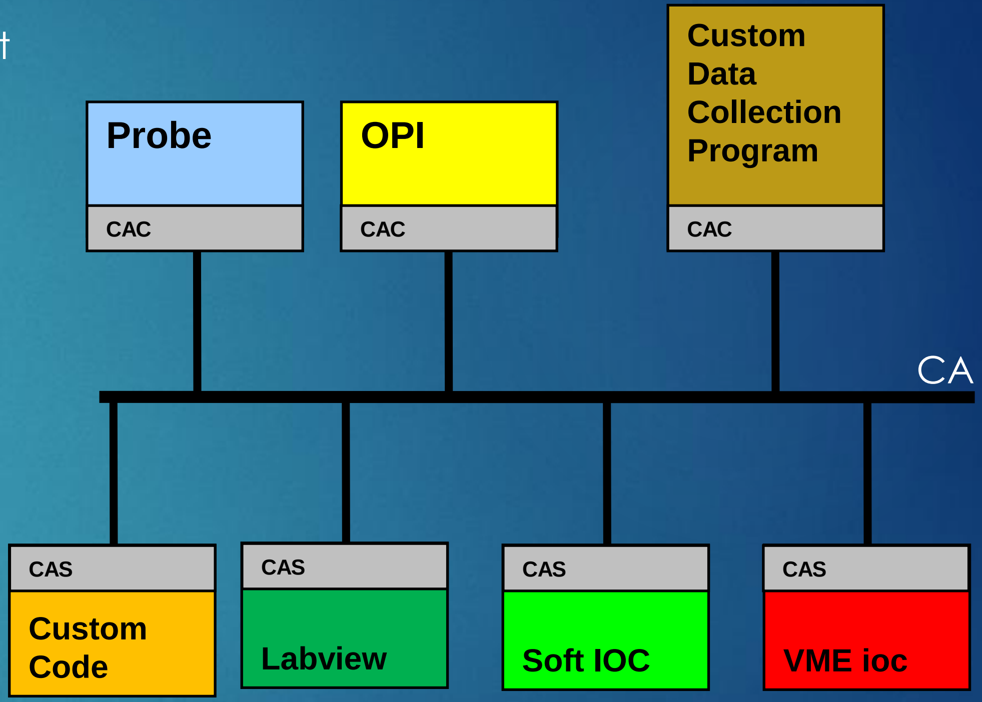
Control System Toolkit

- Any tool/program/application that abides by the Channel Access protocol could be described as "EPICS Compliant"
- EPICS can be viewed as a "toolkit" of EPICS compliant programs. One can select the appropriate tool for their need or develop their own.



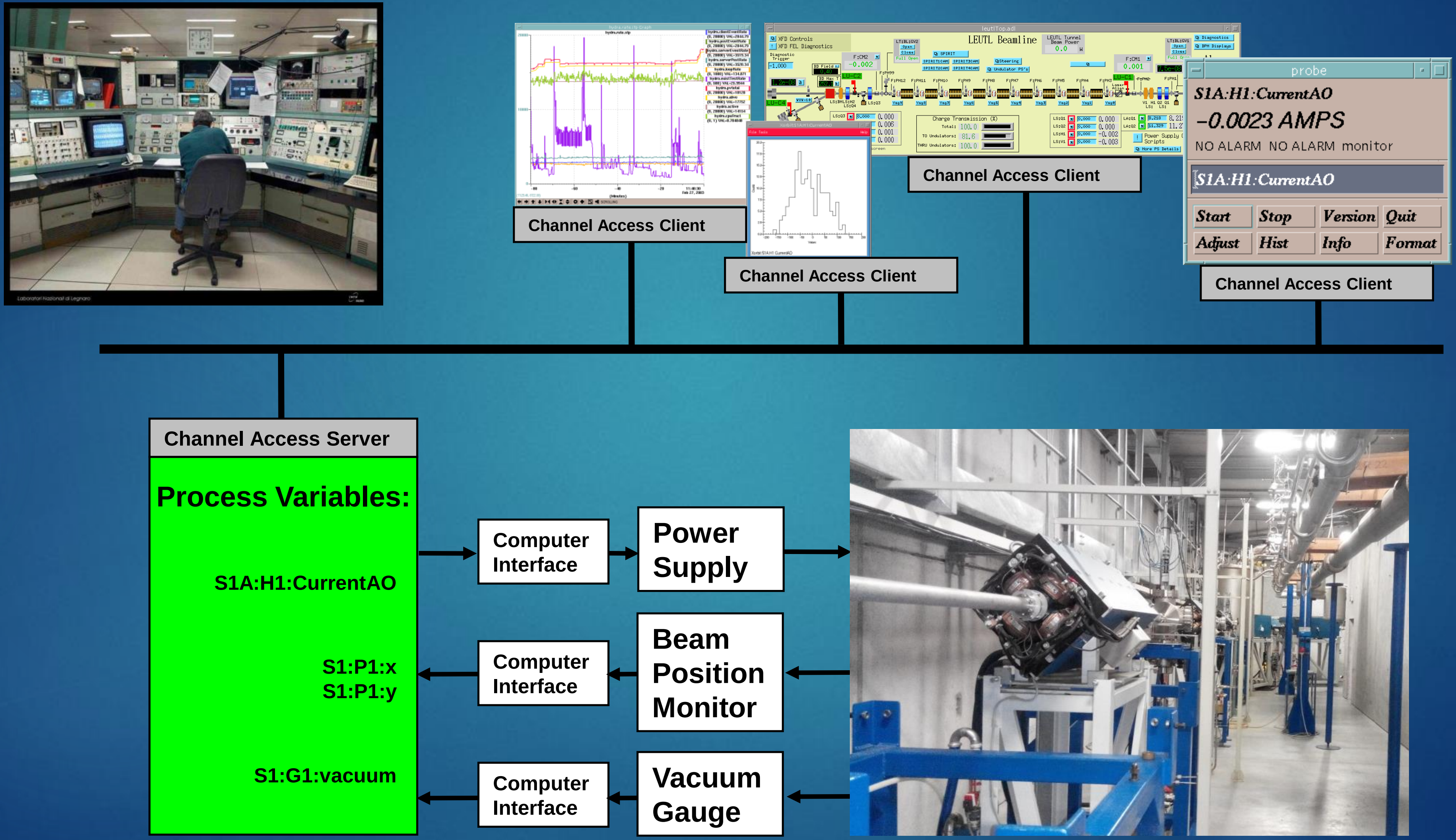
EPICS Toolkit

- Main Features provided by several tools:
  - Remote Control & Monitoring of Technical Equipment
  - Data Conversion/Filtering
  - Closed Loop Control
  - Access Security
  - Equipment Operation Constraints
  - Alarm Detection/Reporting/Logging
  - Data Trending/Archiving/Retrieval/Plotting
  - Automatic Sequencing
  - Mode & Facility Configuration Control (save/restore)
  - Modeling/Simulation
  - Data Acquisition
  - Data Analysis

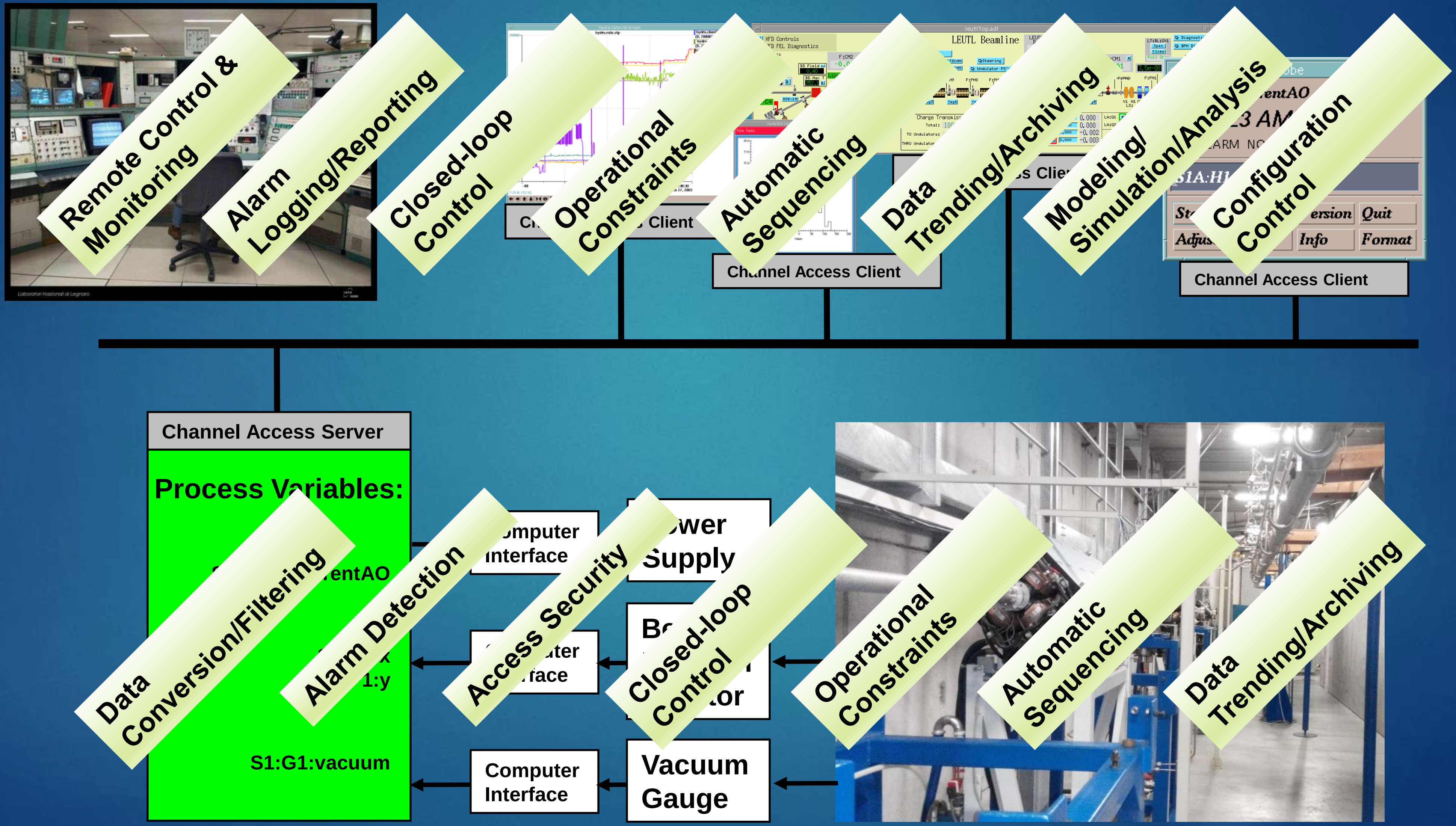


- The set of tools used defines the distributed control system

EPICS Control System - Example



EPICS Control System - Example



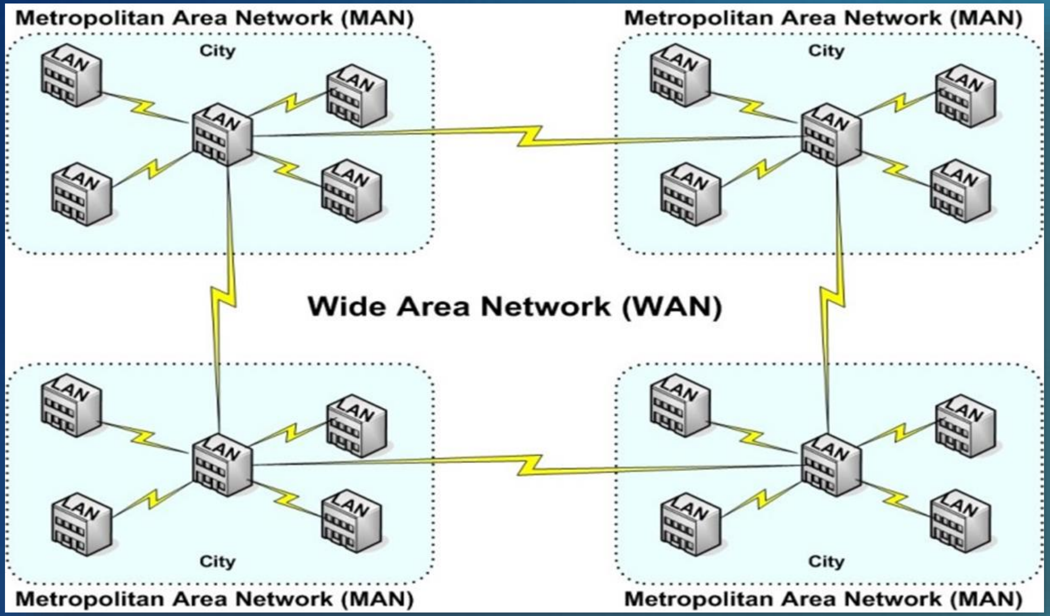
Networking

THE IMPORTANCE OF NETWORK INFRASTRUCTURE IN CONTROL SYSTEMS

Introduction to Network Infrastructure

- **Importance of network infrastructure in control systems**
  - The network infrastructure forms the core communication framework for EPICS
  - A robust network infrastructure ensures that data flows seamlessly between devices, facilitating real-time monitoring and control
- **Key components**
  - network architecture
  - components
  - topologies
  - protocols
  - security measures
- **Importance of a Reliable Network**
  - Any network failure or disruption can lead to significant operational issues, potentially halting experiments or processes.
  - By understanding and implementing robust network infrastructures, we can minimize downtime, enhance security, and ensure the smooth operation of the control systems.

Basic of Network Architecture



- **Local Area Network (LAN)**
  - LANs are networks that cover a small geographic area, typically within a single building or campus.
  - They are used to connect computers and devices within a localized area, enabling high-speed data transfer and resource sharing.
  - In an EPICS control system, LANs are often used to connect devices within a laboratory or facility.
- **Wide Area Network (WAN) and Metropolitan Area Network (MAN)**
  - WANs and MANs cover a larger geographic area, connecting multiple LANs across cities, regions, or even countries.
  - MANs are usually smaller than WANs
  - They facilitate communication and data exchange over long distances, often using leased lines, satellite links.
  - For EPICS systems, WANs might be used to link control systems across different facilities or research centers, allowing centralized or remote monitoring (i.e. IFMIF console at LNL by VPN)

Baselines for designing EPICS network

When designing network architecture for EPICS control systems, several key factors must be considered:

- **Bandwidth**
  - requirements to handle the volume of data traffic.
  - Minimum Bandwidth: **1 Gbps** (Gigabit per second)
  - EPICS systems often handle significant amounts of data, including real-time monitoring and control data. A minimum of 1 Gbps ensures sufficient throughput for most control system applications. Higher bandwidth, such as 10 Gbps, might be required for data-intensive environments.
- **Latency**
  - and response time to ensure real-time control and monitoring.
  - Minimum Latency: **< 1 ms** (millisecond) for LAN; **< 50 ms** for WAN (longer distances)
  - Low latency is important for real-time control and monitoring to ensure timely responses to system events and commands.

Baselines for designing EPICS network

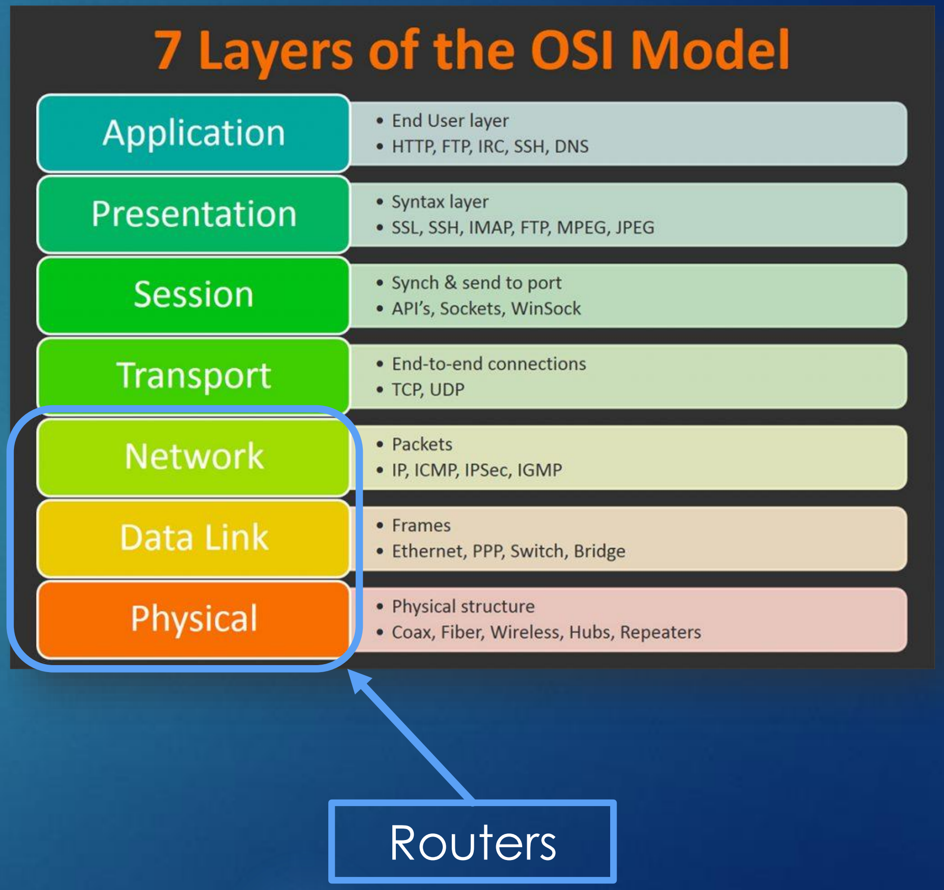
When designing network architecture for EPICS control systems, several key factors must be considered:

- **Scalability**
  - to accommodate future expansion and increased data loads.
  - Number of Devices: the architecture should support scaling up to thousands of devices without significant degradation in performance.
  - Network Segmentation: usage of VLANs or subnetting to manage and segregate network traffic efficiently. (i.e. VLAN to develop, and VLAN to production …)

- **Security**
  - to protect sensitive data and control commands from unauthorized access. (i.e. ACL)
  - Firewall: Implementation of hardware and software firewalls to protect network boundaries
  - VPNs (Virtual Private Networks): For secure remote access and data transfer.
  - Regular updates and patching: Keeping all network devices and software up to date with the latest security patches.

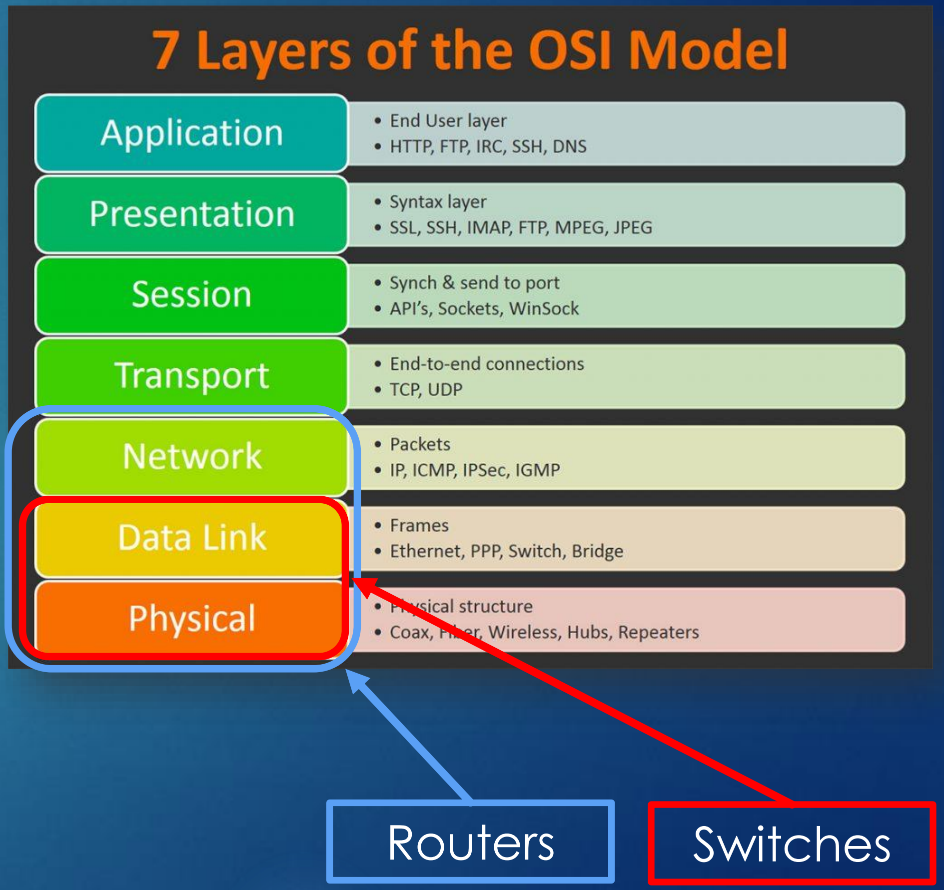
Network Components

- **Routers**
  - Function: forward data packets between computer networks. Think of a router as a traffic director on the internet or within your network at home or work.
  - Routers works at Layer 3 of the OSI model:
    - **Layer 1: Physical Layer**
      for routers, layer 1 ensures that the device can physically connect to the network.
    - **Layer 2: Data Link Layer**
      routers mainly work above this layer but can use MAC addresses to interact with switches.
    - **Layer 3: Network Layer**
      This is where routers primarily operate. The Network Layer uses IP addresses to direct data packets between different networks.
    - NOTE: **Layer 4 and Above**
      while routers operate mainly at Layer 3, they also use information from higher layers, like TCP (Layer 4) and sometimes application-level data (Layer 7), to make more informed decisions about data routing.



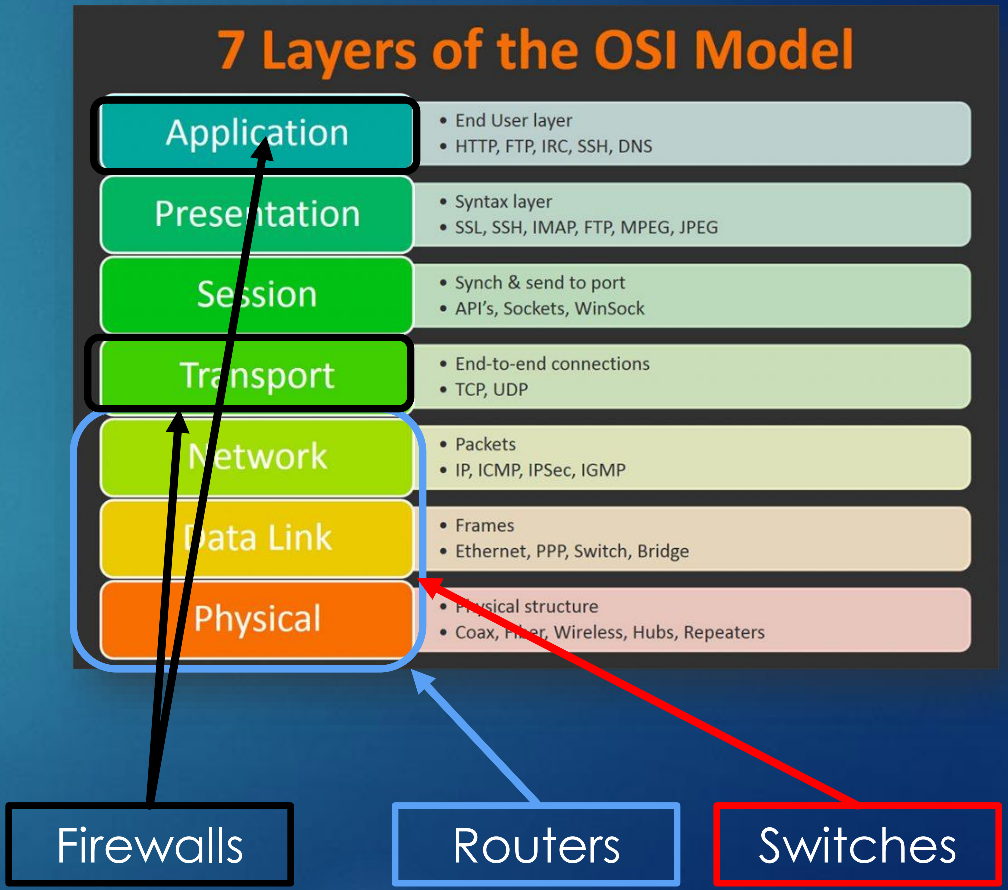
Network Components

- **Switches**
  - Function:
    - Switches are networking devices that connect devices within a single network, such as computers, printers, and servers, enabling them to communicate with each other.
    - Operating at the data link layer (Layer 2) of the OSI model, switches use MAC addresses to forward data to the correct destination.



Network Components

- **Firewalls**
  - Function:
    - Security devices that monitor and control incoming and outgoing network traffic based on predetermined security rules.
    - <u>Types</u>:
      1. hardware-based
      2. software-based
      3. combination of both
    - They operate at various layers of the OSI model, primarily at Layer 4 (transport) and Layer 7 (application)



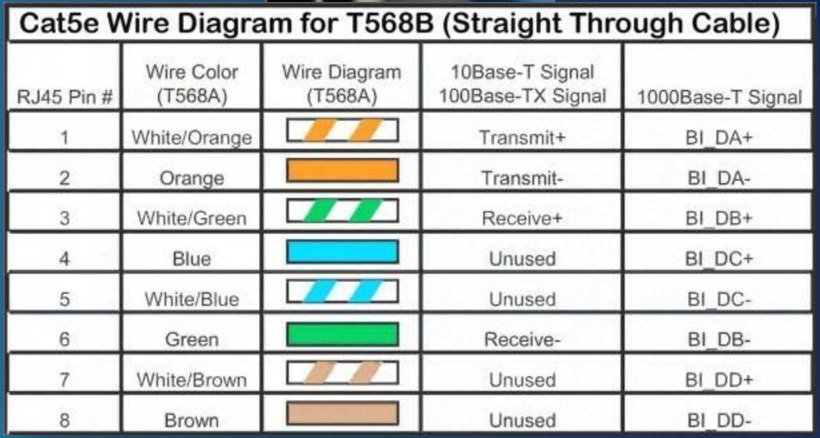
1

2

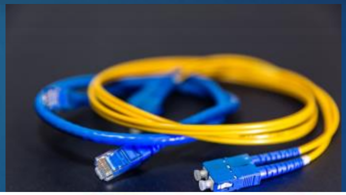
endian firewall community

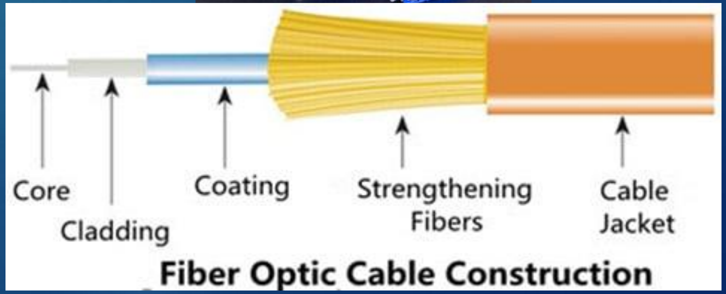
Network Components

- **Network Cabling**
  - Ethernet:
    - Ethernet cables are the most common type of network cabling, used for connecting devices in LANs.
    - They offer a balance of speed, reliability, and cost-effectiveness, supporting speeds up to 10 Gbps and beyond.
    - The most widely used Ethernet cables include
      - **Cat5e - supports speeds up to 1 Gbps**
      - **Cat6 - can handle up to 10 Gbps over shorter distances**
      - **Cat6a - extends Cat6 capability to longer distances**
  - Fiber Optic:
    - Fiber optic cables use light to transmit data, providing much higher bandwidth and longer transmission distances compared to Ethernet.
    - They are ideal for high-speed, high-capacity networks and are often used for backbone connections and data centers


Cat5e Wire Diagram for T568B (Straight Through Cable)

RJ45 Pin #	Wire Color (T568A)	Wire Diagram (T568A)	10Base-T Signal 100Base-TX Signal	1000Base-T Signal
1	White/Orange		Transmit+	BI_DA+
2	Orange		Transmit-	BI_DA-
3	White/Green		Receive+	BI_DB+
4	Blue		Unused	BI_DC+
5	White/Blue		Unused	BI_DC-
6	Green		Receive-	BI_DB-
7	White/Brown		Unused	BI_DD+
8	Brown		Unused	BI_DD-



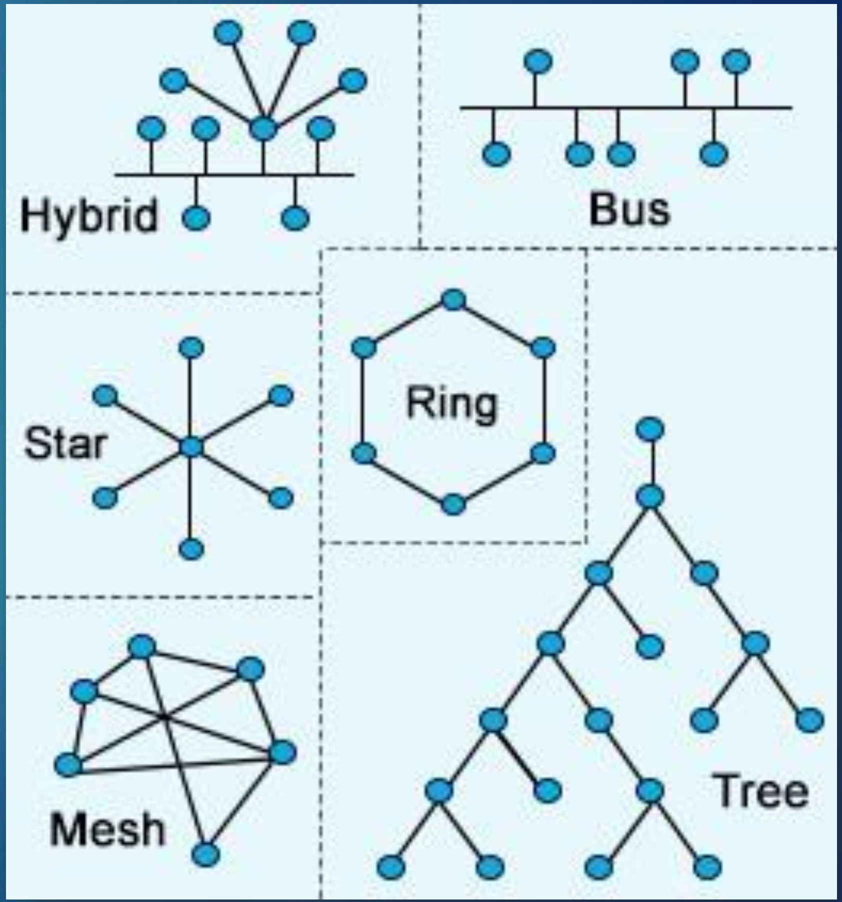


Fiber Optic Cable Construction

Network Topology

- A network topology is the physical and logical arrangement of nodes and connections in a network. Nodes usually include devices such as switches, routers and software with switch and router features

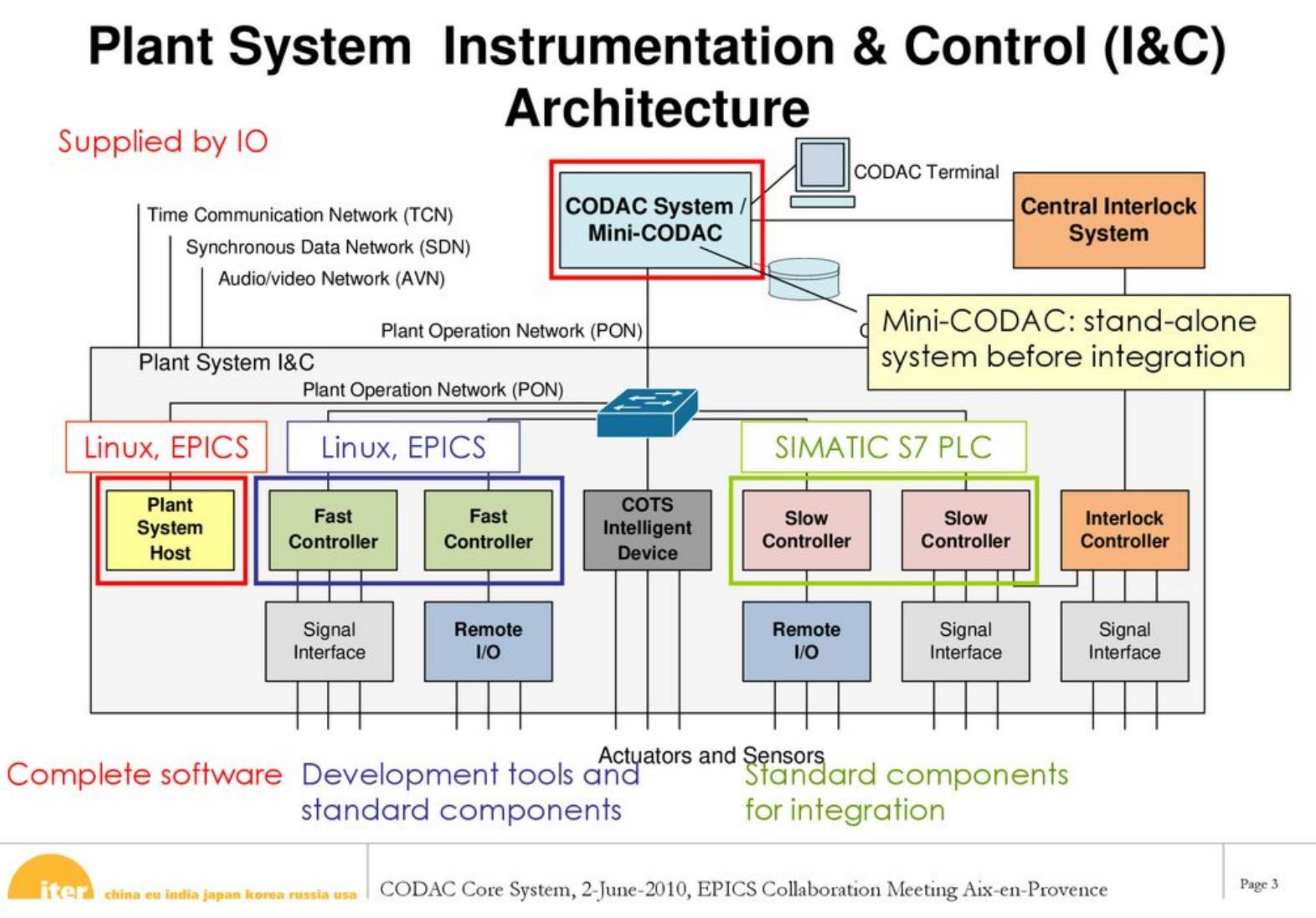
- Most of the cases, control system architectures are based on **Tree type topology**



Network Topology Example: Tree

- **Tree topology is essentially a combination of star and bus topologies.**
- It organizes devices in a hierarchical structure, much like a tree with branches
- Characteristics:
  - the network is divided into levels:
    - The top level is the root, typically a central hub, a gateway, a firewall or a switch.
    - The next level contains intermediate devices (i.e. switches, routers, etc.)
    - The bottom level consists of end devices such as computers, sensors, and controllers.
  - This hierarchical structure allows for organized and scalable network design, making it easier to manage large networks.

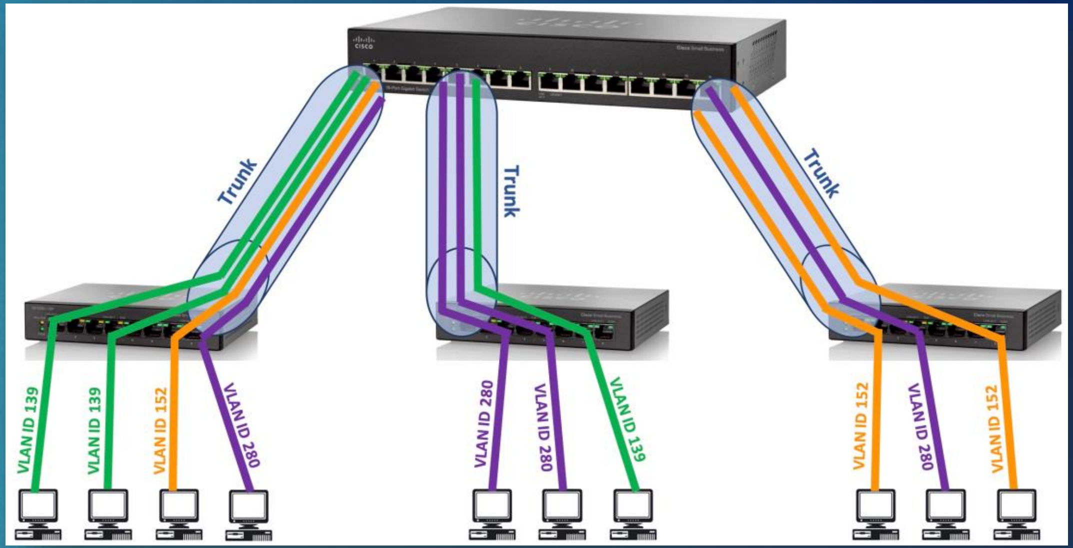
Network Topology Example: Tree



- **Scalable and Expandable:** Easy to add new devices at different levels without disrupting the network.
- **Centralized Management:** The root node helps in managing and controlling the entire network effectively.
- **Fault Isolation:** Problems in one branch do not necessarily affect the entire network, making troubleshooting easier.

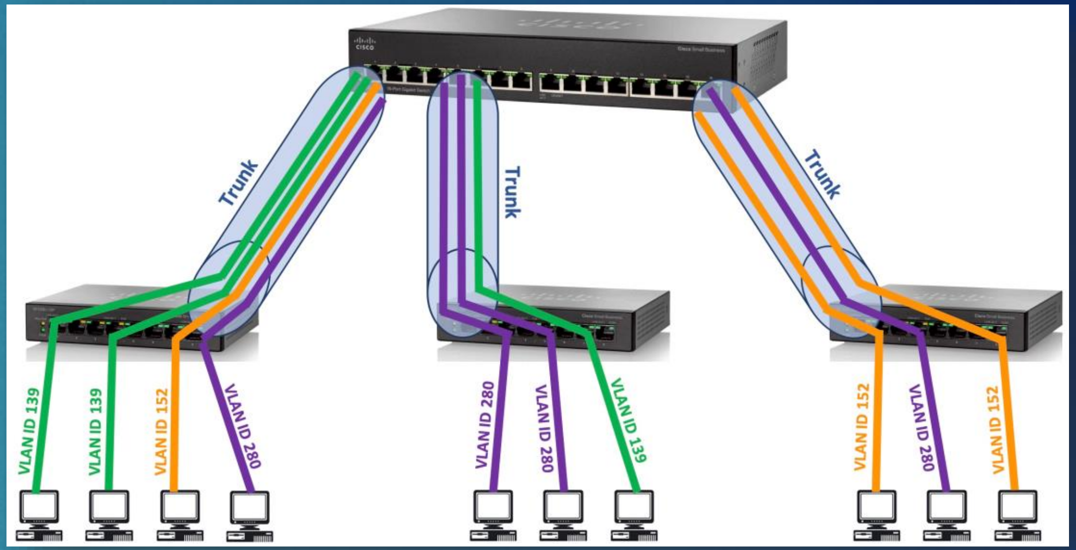
Virtual Local Area Networks

- To improve efficiency and security within a tree topology, we can use **Virtual Local Area Networks** (**VLANs**).
- A VLAN is a network communications domain that is **isolated at the data link layer**. It allows for the creation of logical groups of devices on a network, even if they are not physically located together.
- This mechanism improves security and network efficiency by limiting communication between devices to only those within the same VLAN.
- VLANs are commonly used in enterprise networks to segment traffic and provide better control over network resources.



Virtual Local Area Networks

- **How VLANs work:**
  - VLANs are implemented at the data link layer (Layer 2) of the OSI model.
  - They use tagging mechanisms (IEEE 802.1Q) to identify and separate traffic.
    Each data packet gets a VLAN tag that specifies its VLAN membership. Operating at the data link layer (Layer 2) of the OSI model, switches use MAC addresses to forward data to the correct destination.
  - Network switches and routers then use these tags to forward packets only to devices within the same VLAN, effectively segmenting network traffic.
    - These devices can support multiple VLANs simultaneously.
    - Trunk ports carry traffic for multiple VLANs.
  - Inter-VLAN routing allows communication between different VLANs.



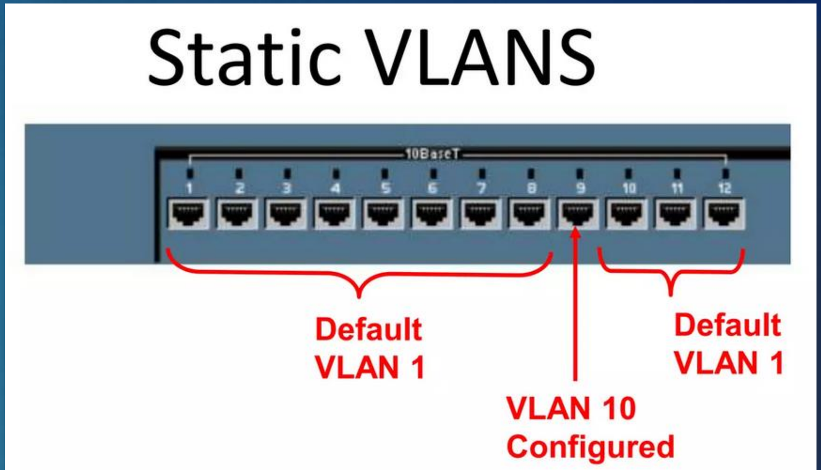
VLANs and Switches

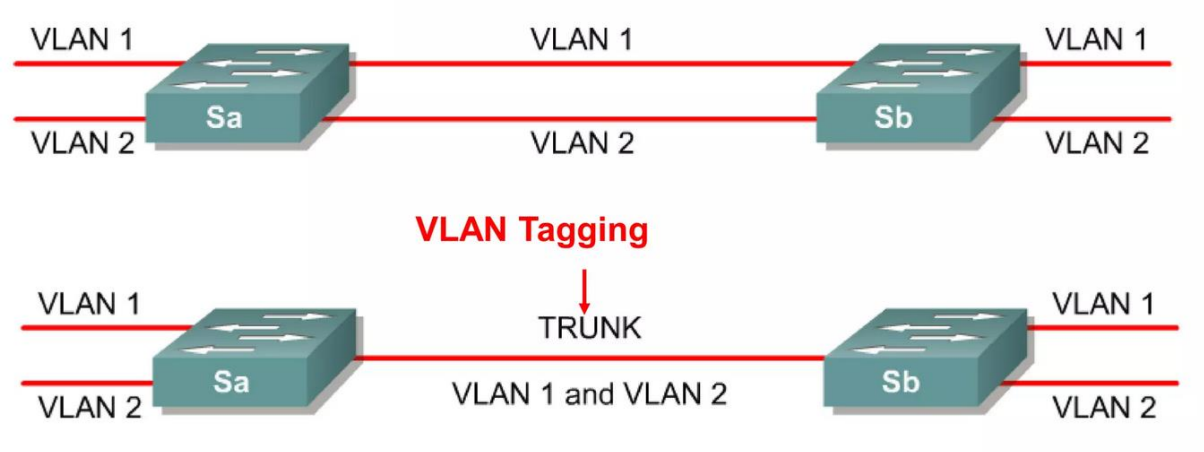
- VLANs are **identified by a number** (range in [1-4094]) which represent the **tag**

- On a VLAN-capable switch, you **assign ports with the appropriate VLAN number**

- The switch then only allows data to be sent between ports with the same VLAN

- **VLAN Tagging is used when a single link needs to carry traffic for more than one VLAN**





Benefit of using VLANs

- **Improved Security:** VLANs can isolate sensitive data from general network traffic, reducing the risk of unauthorized access.

- **Better Performance:** By segmenting the network into smaller broadcast domains, VLANs reduce congestion and improve overall network efficiency.

- **Flexibility:** VLANs allow network administrators to group devices based on function, department, or project, rather than physical location.

VLANs in a CS Architecture

- **Example:**
  - You might have separate VLANs for data acquisition, control signals, and administrative tasks.
  - Each branch of the tree topology can be further divided into VLANs, ensuring that only relevant devices communicate with each other.

- **Implementation:**
  - At the top level (root), the central switch manages VLAN assignments and ensures proper routing of tagged data packets.
  - Intermediate switches at each branch handle VLAN traffic, ensuring that devices on the same VLAN, even if on different branches, can communicate coherently.

Network Protocols

- Communication protocols ensure that information is exchanged accurately and efficiently, allowing for seamless communication between systems regardless of their underlying technologies or architectures
- The principal network protocols used in EPICS control system architectures are:
  - **TCP - Transmission Control Protocol**
  - **UDP - User Datagram Protocol**
  - **SNMP - Simple Network Management Protocol**

Network Protocols: TCP

- **Function**:
  - TCP is the foundational protocol suite for most modern networks, and it is complemented with IP (Internet Protocol)
  - TCP/IP can be hardware-based, software-based, or a combination of both, and they operate at various layers of the OSI model, primarily at Layer 4 (transport) and Layer 7 (application)
  - TCP is a reliable, connection-oriented protocol by establishing a connection between the sender and receiver and providing error checking and flow control. In addition, TCP can recover lost packets through retransmission mechanisms
  - It is used by many applications including HTTP and FTP
- **Advantages**:
  - Reliable data transmission with error correction and retransmission mechanisms
  - Widely supported and compatible with a vast range of devices and applications
- **Usage in EPICS**:
  - TCP/IP is used for critical data transmission in EPICS, such as control commands and status updates, where reliability and accuracy are essential

Network Protocols: UDP

- **Function**:
  - UDP is a simpler and faster protocol compared to TCP. It sends data packets, called datagrams, without establishing a connection.
  - It works at Layer 4 (Transport) of the OSI model
  - There is no error checking or retransmission, making it less reliable but much faster.
  - UDP supports multicast and broadcast communication for efficient data distribution
  - It is commonly used for real-time applications like video streaming and online gaming
- **Advantages**:
  - Low latency and reduced overhead, making it ideal for real-time applications.
  - Efficient for applications where occasional data loss is acceptable.
- **Usage in EPICS**:
  - UDP is used for PV retrieving (via broadcast calls) and applications requiring fast data transmission, such as real-time monitoring and data streaming, where speed is more critical than reliability.

Network Protocols: SNMP

- **Function**:
  - SNMP is an application layer protocol that allows network administrators to monitor and control network devices, such as routers, switches, servers, printers, and more, by sending and receiving information about their status and performance
  - It operates on the application layer (Layer 7) of the OSI model and uses a manager/agent model for communication
  - It operates on UDP port 161/162
  - SNMP can be used to monitor bandwidth usage and network traffic. It can also be used to detect and troubleshoot network issues
- **Advantages**:
  - Provides a standardized method for collecting and organizing network information
  - Enables remote monitoring and management of network devices
- **Usage in EPICS**:
  - SNMP is employed to monitor the health and performance of network components in EPICS systems, ensuring that all devices are operating correctly and efficiently

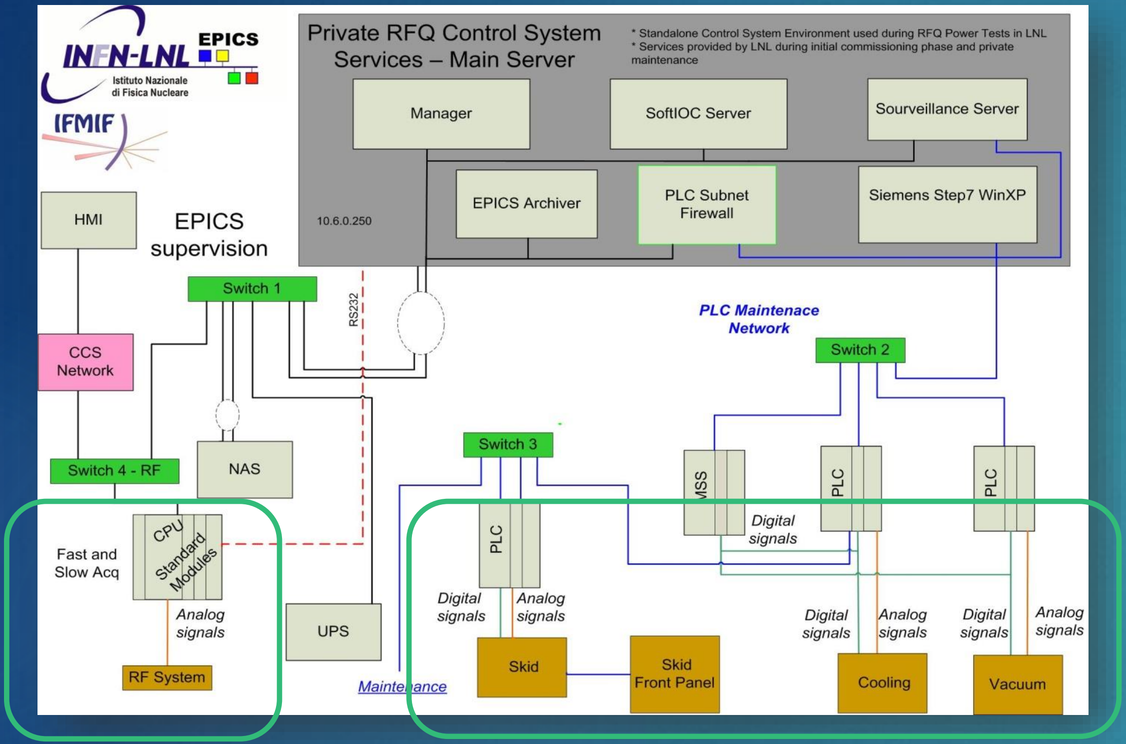
Other Protocols: Telnet, VSTFTP

- **Telnet**:
  - Telnet is a network protocol that allows bidirectional communication between two devices over a virtual terminal connection. It enables users to remotely access and manage devices or systems on a network
  - It operates over TCP/IP and provides a command-line interface for managing network devices
  - Telnet can be used for configuring and troubleshooting network equipment, though it is generally being replaced by more secure alternatives like SSH
  - <u>Telnet is often used to connect to an IOC shell running on a VxWorks box, facilitating remote management and control</u>
- **VSFTP (Very Secoure File Transfer Protocol)**:
  - vsftpd is a secure and fast FTP server for Unix-like systems
  - vsftpd is often used for securely transferring files, including configuration files and updates.
  - In EPICS, <u>vsftpd can be utilized for device setup and configuration due to its robust security features, and for uploading a VxWorks OS at boot time on older CPUs.</u>

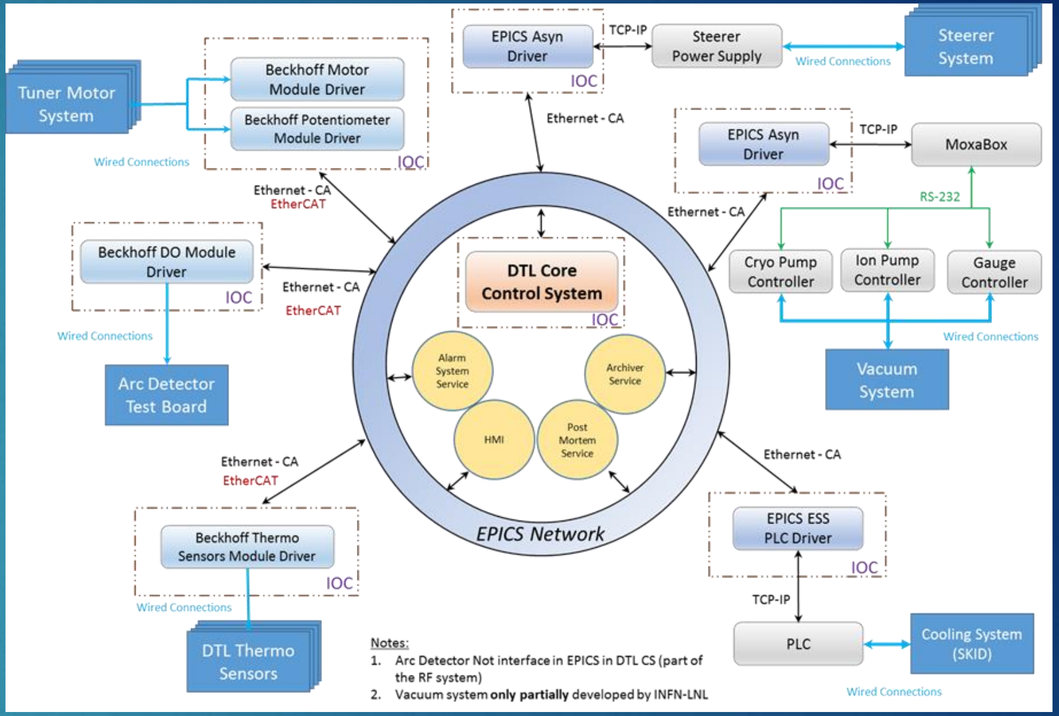
Device Communication

EFFECTIVE DEVICE COMMUNICATION FOR CONTROL SYSTEMS

Device Communication



Communication between the device and the controller to send and acquire data

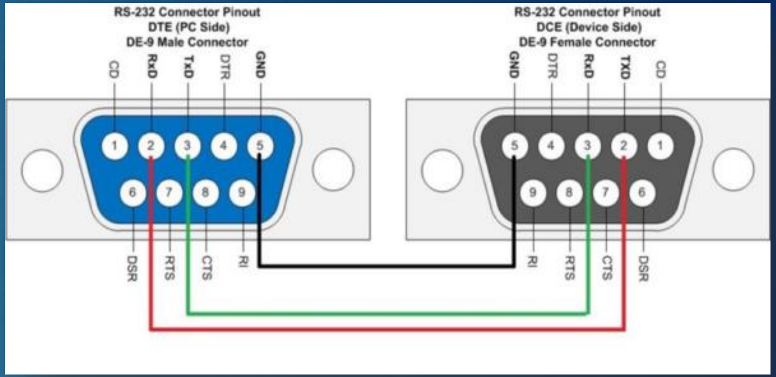


Many ways to exchange data:

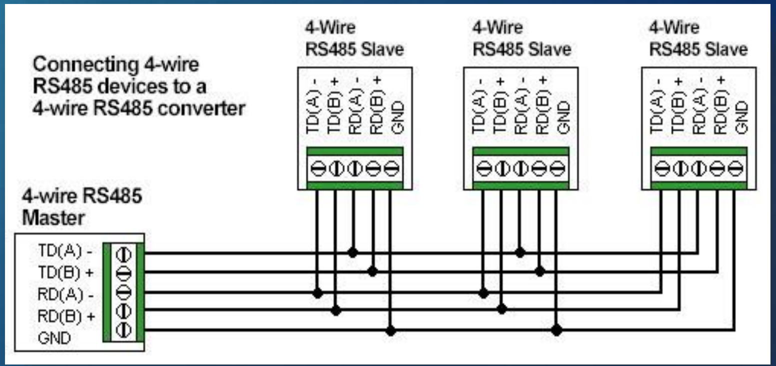
- Different hardware
- Different communication
- Different controllers for the logic

Serial Communication

- **RS232:**
  - Standard for serial communication with low-speed, short-distance data transmission.
    - **Max Distance**: 15 Meters (50 feet)
    - **Max Speed**: 115.2 Kbps
  - Used for connecting peripherals such as sensors and controllers
    - **Max Devices**: Point-to-Point (1 device to 1 device)
    - Common in legacy systems within nuclear facilities



- **RS485:**
  - Supports higher speeds and longer distances than RS232
    - **Max Distance**: 1200 Meters (4000 feet)
    - **Max Speed**: 10 Mbps Kbps
  - Used in industrial environments for connecting multiple devices on the same bus in harsh environment.
    - **Max Devices**: Up to 32 Devices on a single bus (expandable with repeaters)



Ethernet-based Communication

Network-based communication – it uses the same knowledge observed in networking section

	Cat5e	Cat6	Optical Fiber
Max Distance	100 meters	100 meters	Up to 10 kilometers
Max Speed	Up to 1Gbps	Up to 10Gbps	Up to 100Gbps or more
Max Devices	Virtually unlimited within network segments, managed by switches and router		

- **Usage:**
  - Connecting control systems and monitoring devices
  - High-speed data transfer for real-time monitoring
  - Remote access and control of equipment
  - Integration with EPICS for comprehensive control and data acquisition

Interface Cards and Modules

Input/Output (I/O) cards are hardware interfaces that facilitate communication between the central processing unit (CPU) and external devices. They can handle both digital and analog signals

- **Analog I/O:**
  - Handles **continuous signals** (e.g., voltage, current)
  - Typical Input Range: 0-10V, 4-20mA
  - Resolution: 12-bit, 16-bit, 24-bit ADCs
- **Digital I/O:**
  - *Manages **discrete signals** (e.g., on/off states)*
  - Voltage Levels: 5V TTL, 24V industrial standard
- **ADCs (Analog-to-digital Converters):**
  - Convert analog signals to digital data for processing
  - Resolution: 12-bit to 24-bit
  - Sampling Rate: Up to 1 MSPS (Mega Samples Per Second)
- **DACs (Digital-to-Analog Converters):**
  - Convert digital data to analog signals for control
  - Resolution: 12-bit to 16-bit
  - Output Range: 0-10V, 4-20mA

Slow Controls: PLC

PLC (**Programmable Logic Controller**) is a type of digital computer that is specifically designed to control and automate machinery or industrial processes. It is commonly used in manufacturing plants, factories, and other industrial settings.

- ***Input/output (I/O) Capabilities****:*
  - *Digital I/O: Typically 24V DC, 120V AC, 240V AC*
  - *Analog I/O: 0-10V, 4-20mA*
  - *Modular I/O Expansion: Allows for scalability*
- **Processing Speed:**
  - Scan Time: 1 ms to 10 ms per 1K instructions
- **Memory:**
  - Program Memory: Ranges from a few KB to several MB
  - Data Memory: For storing variables and temporary data
- **Communication Protocols:**
  - Ethernet/IP, Modbus, Profibus, DeviceNet
- **Environments Specs:**
  - Operating Temperature: 0°C to 60°C
  - Shock and Vibration: Designed to withstand harsh industrial environments
- **Features:**
  - Robust and reliable operation
  - Programmable for various control tasks
  - Redundancy and failover capabilities for critical applications
  - Integrated HMI (Human-Machine Interface) for user interaction

Fast Controls: MicroTCA

Micro Telecommunications Computing Architecture (MicroTCA or μTCA) is a modular, open standard developed by the PCI Industrial Computer Manufacturers Group (PICMG) for building high-performance, reliable, and scalable computing and communication systems.

- ***Architecture***:
  - *Based on AdvancedTCA (ATCA) standards, but more compact*
  - Supports multiple Advanced Mezzanine Cards (AMCs) in a single chassis
- **Processing Power:**
  - Supports multi-core processors and high-speed interconnects
  - Typical data throughput: Up to 40 Gbps per channel
- **Latency:**
  - Ultra-low latency, typically in the microsecond range
- **Redundancy and Reliability:**
  - Hot-swappable modules for high availability
- **Environments Specs:**
  - Operating Temperature: -40°C to 85°C
  - Designed for high vibration and shock environments
- **Features:**
  - Modular and scalable architecture for flexible system design
  - High-speed data acquisition and processing
  - Integrated management and monitoring capabilities
  - Redundancy and failover mechanisms for critical applications

Field Buses

UNDERSTANDING AND IMPLEMENTING FIELD BUSES FOR EFFICIENT CONTROL

Introduction to Field Buses

- **What are Field Buses**
  - **Definition:** Field Buses are industrial network systems used for real-time distributed control
  - **Purpose:** They enable communication between various devices such as sensors, actuators, and controllers in a control system, allowing for efficient communication and data exchange between devices on the field within industrial processes
  - **Key Aspects**:
    - Real-time Communication
    - Decentralized Control
    - Interoperability
- **Roles of Field Buses in Control System**
  - Field Buses streamline data exchange and control commands, increasing the efficiency and reliability of control systems.

Common Field Buses

- **CAN Bus (Controller Area Network)**
  - Widely used in automotive and industrial applications
  - Supports real-time control and high data integrity
- **Profibus (Process Field Bus)**
  - Common in process automation and factory automation
  - Offers high-speed communication and extensive diagnostic capabilities
- **Modbus**
  - Simple and open protocol used in various industrial environments
  - Supports communication over serial and Ethernet networks

CAN Bus

- **Features**
  - CAN bus is a standard for device communication without a host computer
  - It was developed by Bosch in the 1980s
  - Real-time communication with high reliability
  - It employs message-based communication with prioritization and error detection
  - CAN bus supports data rates up to 1 Mbps and the maximum length of a CAN bus network is typically 40 meters

- **Integration with EPICS**
  - EPICS modules are available for integrating CAN Bus networks.

Profibus

- **Features**
  - High-speed communication and extensive diagnostic functions
  - Supports a wide range of devices and configurations and it can handle multiple devices on a single network
  - Provides reliable and robust communication in harsh industrial environments
  - Offers flexibility with different transmission speeds and cable types
  - Enables efficient diagnostics and monitoring of connected devices
- **Integration with EPICS**
  - EPICS supports Profibus through various hardware interfaces and software drivers.

Modbus

- **Features**
  - Simple and open protocol with wide adoption
  - Developed by Modicon in 1979 for PLCs
  - Supports master-slave architecture with up to 247 slaves
  - Uses RS-232, RS-485 or Ethernet physical layers
  - Has simple and efficient message structure and it supports multiple data types and functions codes
  - Has become an industry standard for industrial automation systems, and ontinues to evolve with new variants such as Modbus TCP/IP

- **Integration with EPICS**
  - EPICS supports Modbus for both serial and TCP/IP communication.

Comparison of Field Buses

Feature	CAN bus	PROFIBUS	Modbus
Communication Type	Peer-to-peer	Master-slave	Master-slave
Data Transmission	Message-oriented	Field-oriented	Field-oriented
Speed	Up to 1 Mbps (Classical CAN)	Up to 12 Mbps (PROFIBUS DP)	Up to 115.2 kbps (Modbus RTU)
Topology	Bus, star, tree	Bus, star	Bus, star
Medium	Twisted pair, fiber optics	Twisted pair, fiber optics	Twisted pair, fiber optics, Ethernet (for Modbus TCP)

Comparison of Field Buses

Feature	CAN bus	PROFIBUS	Modbus
Max. Nodes	Up to 110	Up to 126	Up to 247 (Modbus RTU), virtually unlimited (Modbus TCP)
Distance	Up to 40 meters at 1 Mbps	Up to 100-1200 meters	Up to 1200 meters (Modbus RTU)
Error Handling	Robust error detection & correction	Comprehensive error-checking	Simple CRC error-checking
Network Management	Device-oriented, CANopen adds higher-layer protocols	Centralized configuration and diagnostics	Simple, minimal configuration needed

IT Architecture for EPICS

BUILDING SCALABLE AND ROBUST INFRASTRUCTURE

Overview of IT Architecture for Control Systems

- **Components of IT Architecture:**
  - Servers
  - Data Storage Solutions
  - Network Infrastructure
  - Database Systems

- **Importance of Scalable and Robust IT Infrastructure**
  - Ensures reliable and efficient operation of control systems
  - Facilitates easy expansion and adaptability to future requirements

Server Infrastructure

- **Type of Servers:**
  - **Web Servers:** Designed to serve HTTP Content
  - **Application Servers:** Run application software and services; can also serve HTTP Content but is not limited to just HTTP (i.e. Remote Method Invocation (RMI), Remote Procedure Call (RPC))
  - **Database Servers:** Store and manage databases

- **Key Concept: High Availability** Redundant systems and failover mechanisms to ensure continuous operation

Server Infrastructure

To avoid BareMetal servers' proliferation:

- **Virtualization Technology**: Running multiple virtual machines on a single physical server
  - **Possible tools**: VMware, KVM, Hyper-V
  - **Benefits**: Resource optimization, isolation, ease of management
- **Containerization Technology**: Packaging applications with their dependencies in isolated containers
  - **Possible tools**: Docker, Podman, Kubernetes
  - **Benefits**: Portability, scalability, fast deployment

Data Storage Solutions

- **Type of storage:**
  - **SAN (Storage area Network)**
    - High-speed network that provides block-level storage
  - **NAS (Network Attached Storage)**
    - File-level storage connected to a network
  - **Cloud Storage**
    - Scalable, on-demand storage managed by third-party providers

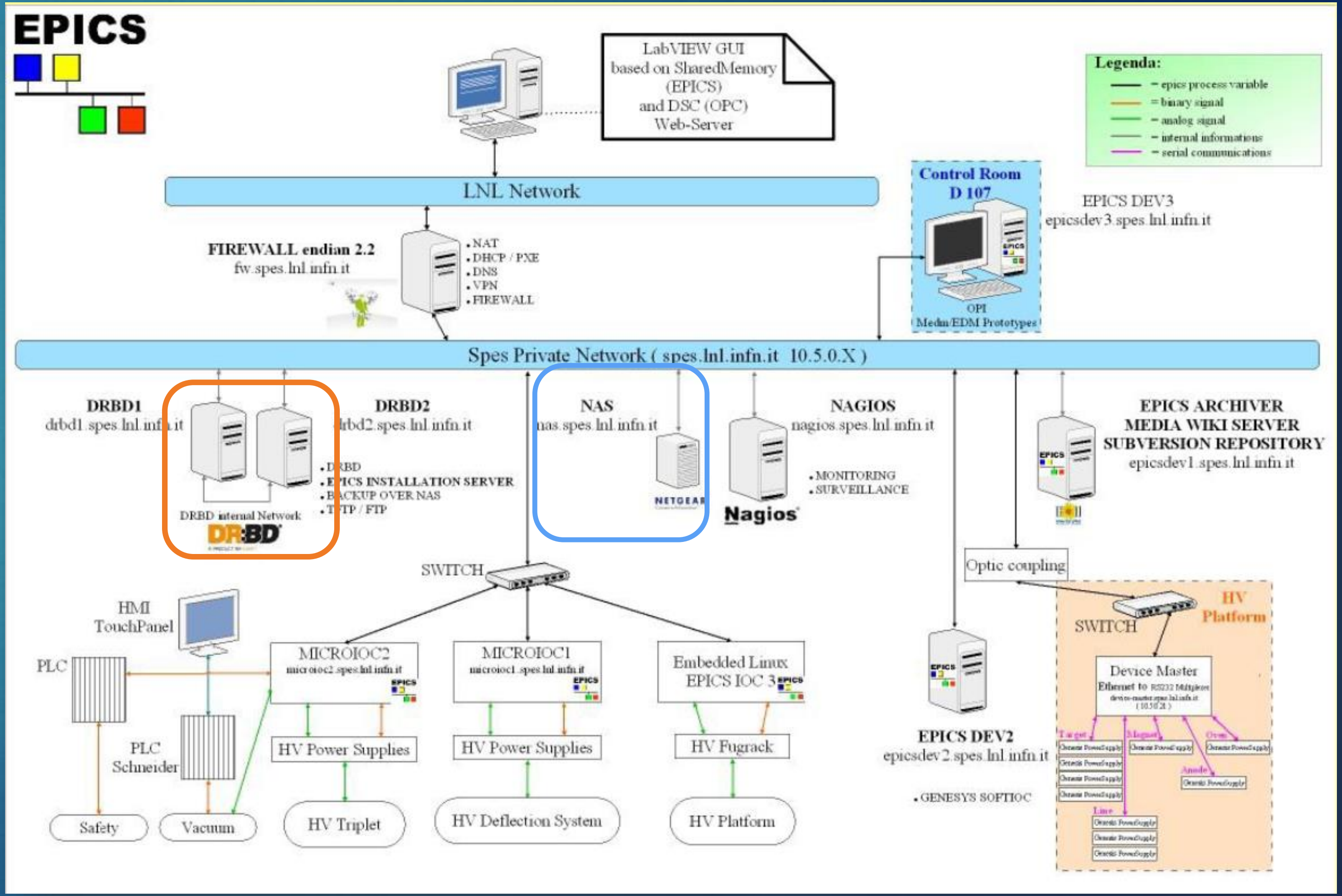
SAN

NAS

Cloud

Data Redundancy and Backup Strategy

- **Data Redundancy**:
  - Ensures data is duplicated across multiple systems to prevent loss
- **Backup Strategy**:
  - Regular backups: Daily, weekly, monthly
  - Off-site storage: Protects against local disasters
  - Disaster recovery plans: Procedures to restore systems and data



Database Management

Type of Databases:

- **SQL / RDB Databases**: Relational databases that use structured query language (SQL) to manage data.
  - **Examples**: MySQL, PopstgreSQL, Oracle, Microsoft SQL Server
  - **Advantages**: Strong consistency, complex queries, and transactions
  - **Use Cases:** Suitable for applications requiring complex queries and transactions
- **NoSQL / NoRDB Databases**: Non-relational databases designed to handle large volumes of unstructured or semi-structured data.
  - **Examples**: MongoDB, Cassandra, CouchDB, Redis
  - **Advantages**: High scalability, flexible schema design, better performance for specific use cases
  - **Use Cases:** Ideal for applications requiring high availability and scalability, such as big data analytics that require flexible storage solution

Note: different EPICS applications and tools can require RDB or NoRDB databases. Therefore, their knowledge is important to proper configure the services

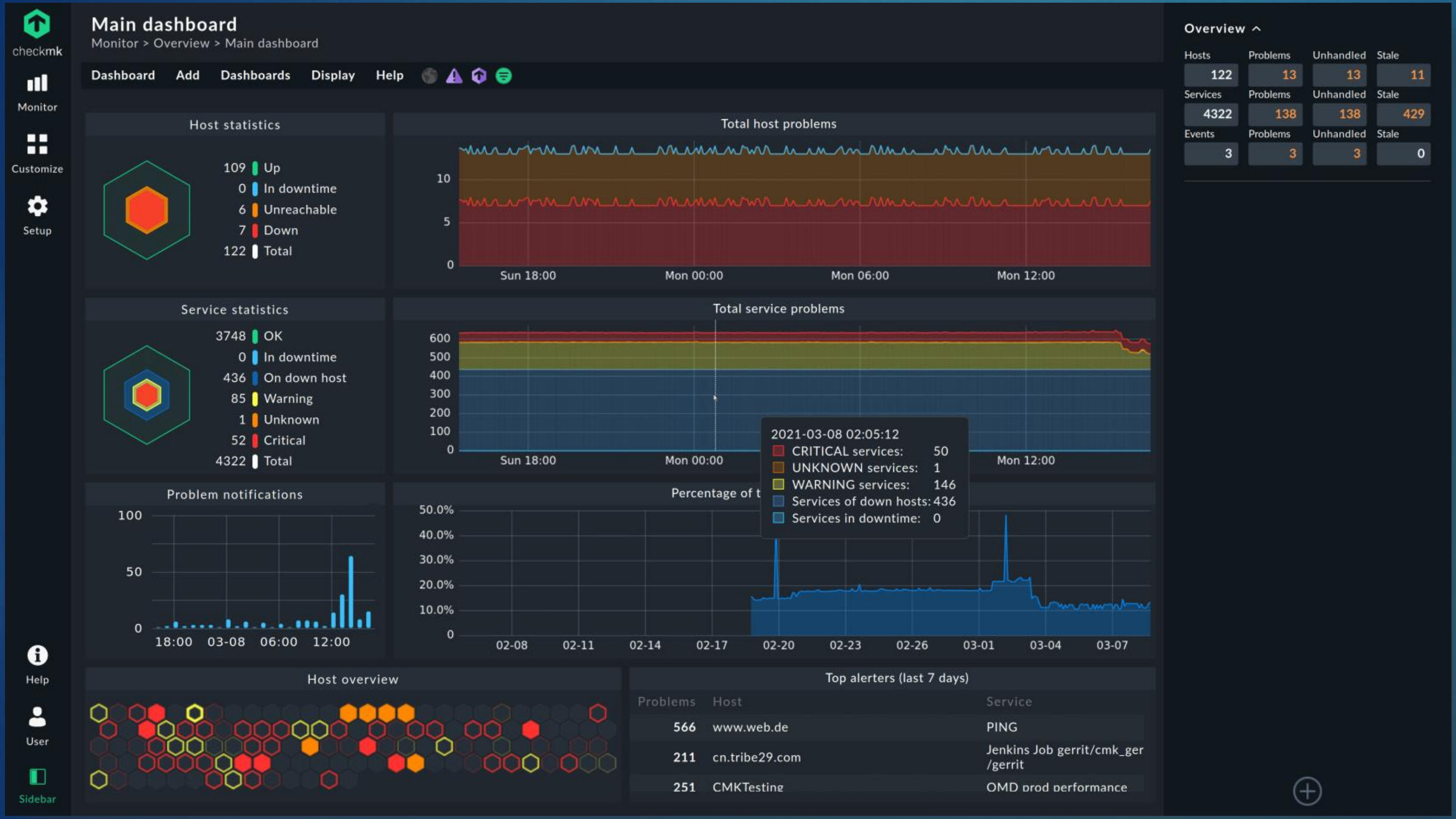
Database Architecture

- **Single Server:**
  - Simple setup with all database services running on a single
  - **Pros:** Easy to manage, cost-effective for small applications
  - **Cons:** Limited scalability and fault tolerance
- **Clustered Databases**:
  - Multiple servers work together to provide database services.
  - **Pros:** High availability, fault tolerance, and better performance
  - **Cons:** More complex to manage and set up
- **Backup and Recovery**:
  - Regular backups to ensure data can be restored in case of failure
  - **Types of Backups:** Full, incremental, differential
  - Automated backup solutions and regular testing of backup procedures are essential to ensure data integrity and quick recovery

Software Tools and Middleware

ENHANCING NETWORK MANAGEMENT, CONTROL, AND DATA PROCESSING

Tools for Network Management



- Network management software tools are key services for IT administrators to monitor, manage, and optimize network performance
- These tools provide capabilities to oversee network operations, identify and troubleshoot issues, ensure security, and maintain network infrastructure

Tools for Network Management

- **Network Monitoring Tools:**
  - ***Wireshark***:
    - *Captures and analyzes network traffic*
    - Helps diagnose network issues and ensure data integrity
  - **Nagios / CheckMK:**
    - Monitors network and system performance
    - Alerts administrators to potential problems and provides detailed reports

- **Configuration Management Tools:**
  - ***Ansible***:
    - *Based on Python, automates configuration management and application deployment with agent-less approach*
    - Uses simple YAML files for defining configurations
  - **Puppet:**
    - Manages configuration of systems through declarative code
    - Ensures consistency across various environments

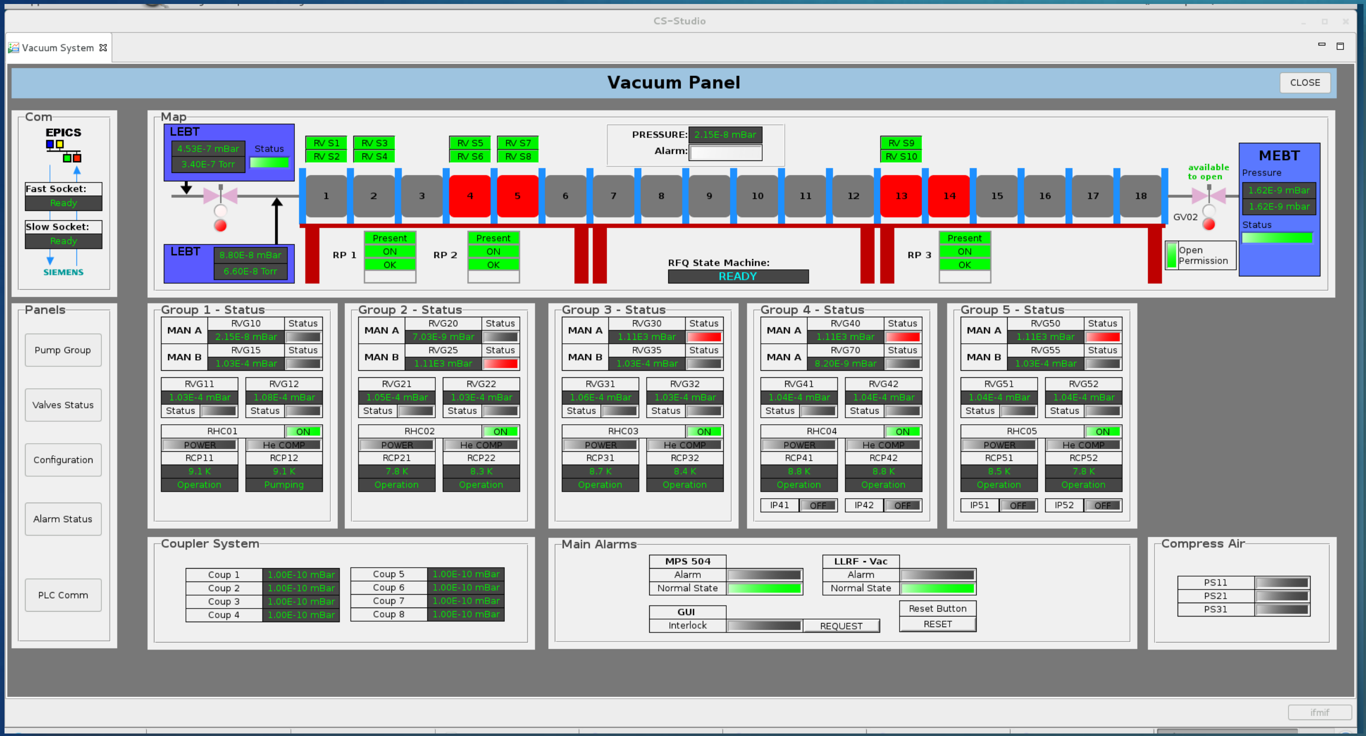
Middleware Solutions

- **Role of Middleware in Control Systems:**
  - Acts as an intermediary layer between software applications and hardware.
  - Facilitates communication, data exchange, and interoperability.
- ***Example of Middleware***:
  - **OPC UA (OLE for Process Control Unified Architecture):**
    - Provides a standard for secure and reliable data exchange.
    - Supports real-time control, data modeling, and event handling.

Human-Machine Interface in EPICS

CONTROL THROUGH EFFECTIVE USER INTERFACES

Role of HMI in Control Systems



- **Importance of user-Friendly Interface:**
  - Facilitate operator interaction with control systems
  - Enhance situational awareness and decision-making
  - Improve efficiency and reduce errors in system operation
- ***Functions of HMI:***
  - Display real-time data and system status
  - Provide control and command interfaces
  - Alarm and event management

Designing Effective HMIs

- **Best Practices for HMI Design:**
  - Keep the interface **simple and intuitive**
  - Use **consistent design elements and color schemes**
  - **Prioritize critical information** and minimize clutter
  - Provide **clear and concise feedback to user actions**
  - Ensure the **interface** is **responsive** and performs well under load

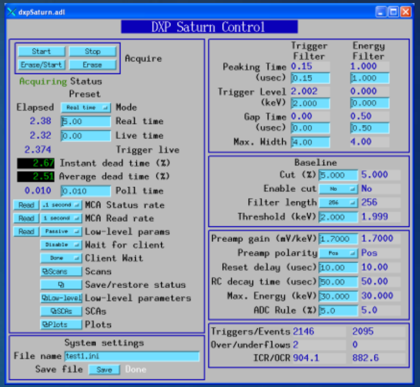
  - Engage the end user (scientist, operator, etc.) in the design process

Designing Effective HMIs

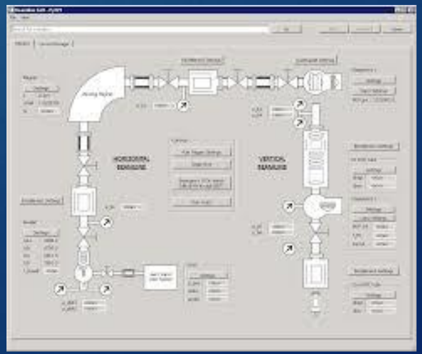
- ***Tools and Software for HMI Development in EPICS*:**
  - **CSS (Control System Studio) and CSS Phoebus**: comprehensive suite for creating EPICS HMIs (more than a simple GUI tool)
  - **MEDM (Motif Editor and Display Manager)**: An older tool, but still widely used for creating EPICS displays
  - **PyQt and PyDM**: Modern frameworks for developing cross-platform HMIs using QT libraries

CSS

MEDM



PyDM



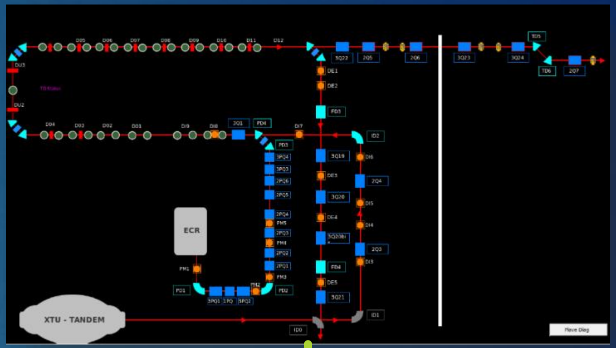
HMI Integration with EPICS

- **Techniques for Integrating HMI with EPICS:**
  - Use EPICS communication protocols for real-time data communication
    - Principal tools (i.e. Phobus) already integrate EPICS Channel Access and PVAccess
  - Leverage EPICS tools like CSS, Phoebus or MEDM for seamless integration
    - Use MEDM only if strictly required, prefer last technological solutions like Phoebus for brand new projects.
  - Ensure secure and reliable communication between HMI and EPICS
    - GUIs tools interact with EPICS servers available in the control system architecture. Networking is a critical aspect also in this topic

HMI Integration with EPICS

- ***Examples of Effective HMI Implementations:***
  - **Synoptic Displays:**
    - Show a high-level overview of the system
    - Allow operators to drill down into detailed views
  - **Alarm Management Interfaces:**
    - Provide real-time alerts and status updates
    - Enable quick identification and resolution of issues
  - **Control Panels:**
    - Offer intuitive control and command interfaces
    - Include interactive elements for real-time adjustments

High-level overview



Detailed view (mag. elem)

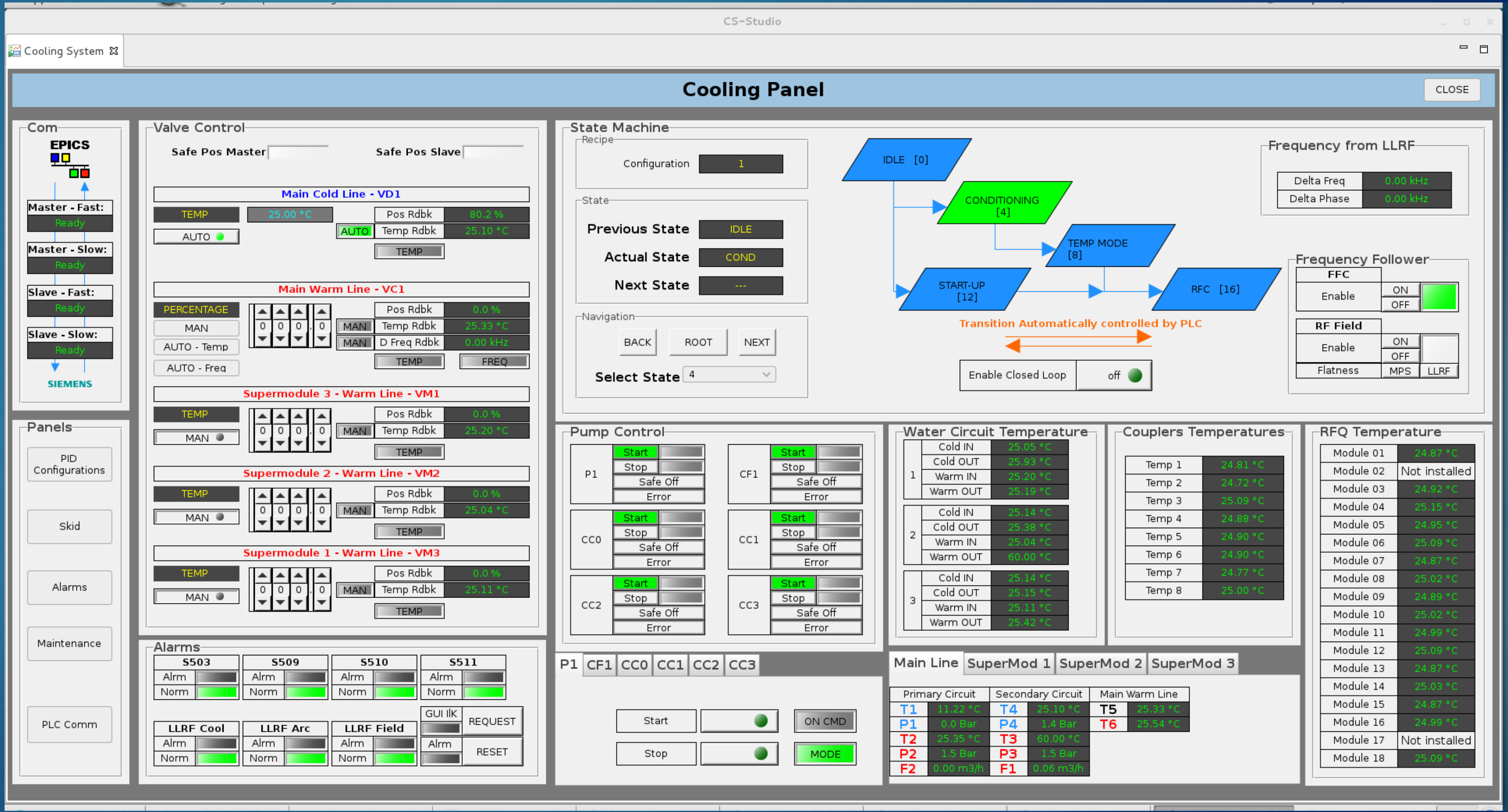
Detailed view (alternative)

Importance of User-Friendly Interface

- **Enhance operator efficiency and accuracy**
- **Reduce training time for new users**
- **Increase system reliability by reducing operators' errors**
- **Facilitate faster decision-making and response time**

Example of Effective HMI implementations

- **IFMIF Project**
- RFQ Apparatus
- Cooling System

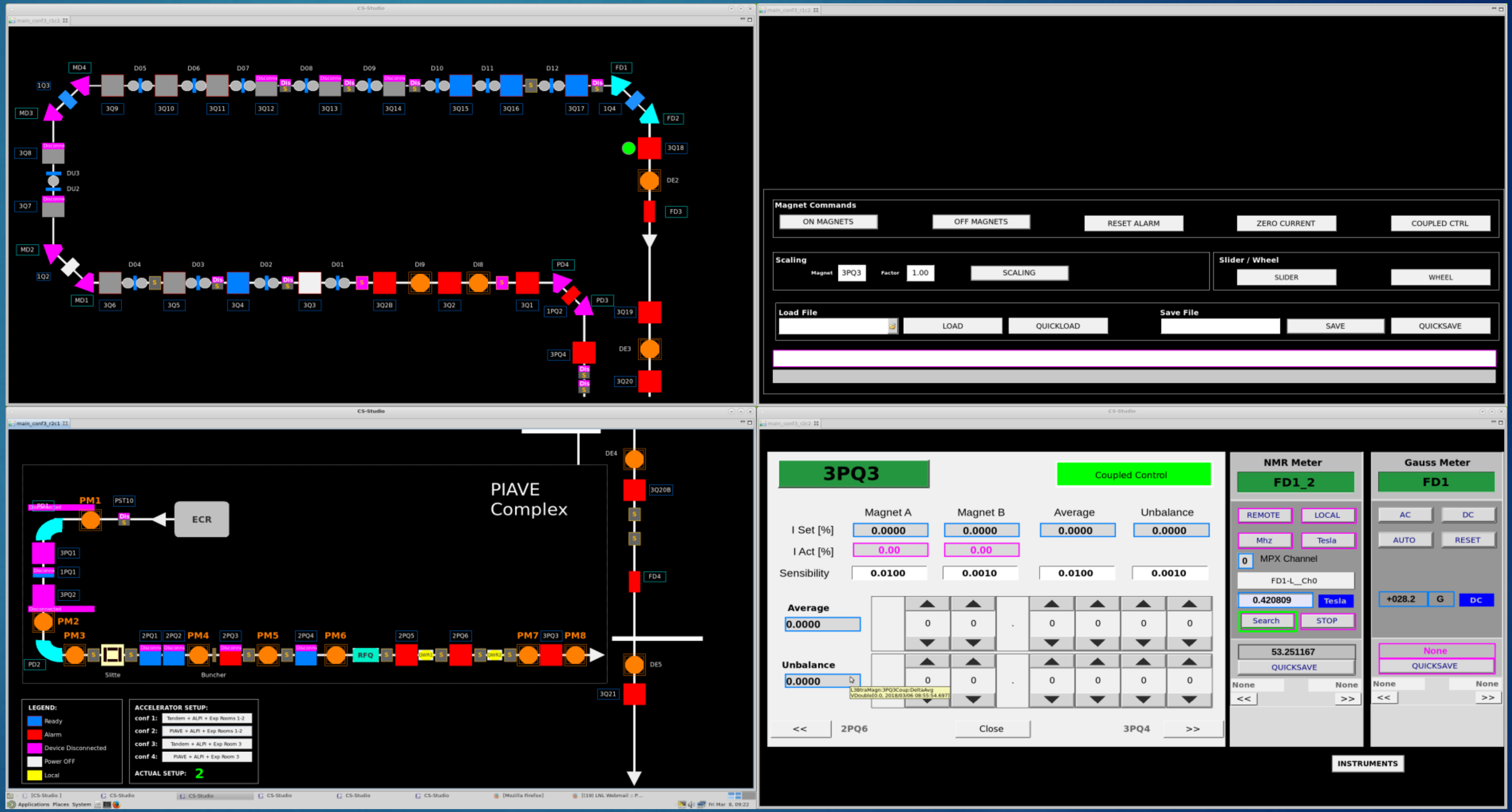


Example of Effective HMI implementations

- **ESS Project**
- DTL Apparatus
- Steering System

Example of Effective HMI implementations

- **SPES Project**
- ALPI/PIAVE Accelerator Lines
- Magnetic Beam Transport System



Documentation in EPICS

ENSURING QUALITY AND CONSISTENCY

… AKA THE HARDEST PART OF THE JOB

Importance of Documentation

- **User Manual:**
  - Guides for end-users to operate the system
  - Include step-by-step instructions and troubleshooting tips
- ***Technical Documentation***:
  - Detailed technical specifications and system architecture
  - Targeted at developers and engineers for system maintenance and upgrades
- **Operational Procedure:**
  - Standard operating procedures (SOPs) for daily operations
  - Include safety protocols and emergency procedures
- **Wiki:**
  - Collaborative platform for storing and sharing documentation
  - Allows for continuous updates and community contributions

Strategies for Keeping Documentation Updated

- Regular reviews and updates
- Assign responsibility for documentation management
- Implement review and approval processes
- Encourage user feedback and contributions
- Don’t forget to comment the code, but don’t use it as principal (and exhaustive) documentation

Strategies for Keeping Documentation Updated

- **Tools for Managing Documentation:**
  - Document management systems
    - Confluence
    - SharePoint
  - Automated documentation tools
    - Doxygen

Versioning with Git

- **Benefits of Using Git for Documentation Versioning:**
  - Tracks changes to documentation over time
  - Facilitates collaboration among multiple contributors
  - Allows for rollbacks to previous versions if needed
  - Provides a clear history of changes and updates
- **Key Features of Git:**
  - Distributed version control system
  - Branching and merging for managing different versions
  - Commit history for tracking changes
  - Git repositories (e.g., GitHub, GitLab, Bitbucket) for hosting and sharing documentation

Conclusion about Documentation

- **Documentation is crucial for EPICS control systems**
  - For end users
  - For maintainers
- **Effective documentation ensures usability and maintenance**
- **Using tools like Git helps manage and maintain documentation**

Thank You for Your Attention

CREDENTIALS

THANKS TO THE EPICS COMMUNITY FOR ALL THE MATERIAL PROVIDED FOR THIS PRESENTATION

HTTPS://EPICS.ANL.GOV