
# Control System & Control with AI

阎映炳 (Yingbing YAN)

SARI-CAS

# Outline

1. Control System Overview
2. Hardware Platform, Interface, Protocol
3. EPICS Basic, Modules, Applications
4. Control Infrastructure, Services, HLA
5. Machine Protection, Timing, Feedback
6. Control with AI

# What is an accelerator control system?

All particle accelerators depend on control systems to **integrate** different **devices** and the autonomous **controllers** that are distributed throughout the facility into **one coherent infrastructure**. The control system provides an **abstraction layer** between hardware and the operators, and creates the **environment** that allows scientists to carry out their experiments. It also enables the technical support groups to compare previous data with current one in order to enhance the performance of their systems.

--- Elke Zimoch, Paul Scherrer Institute

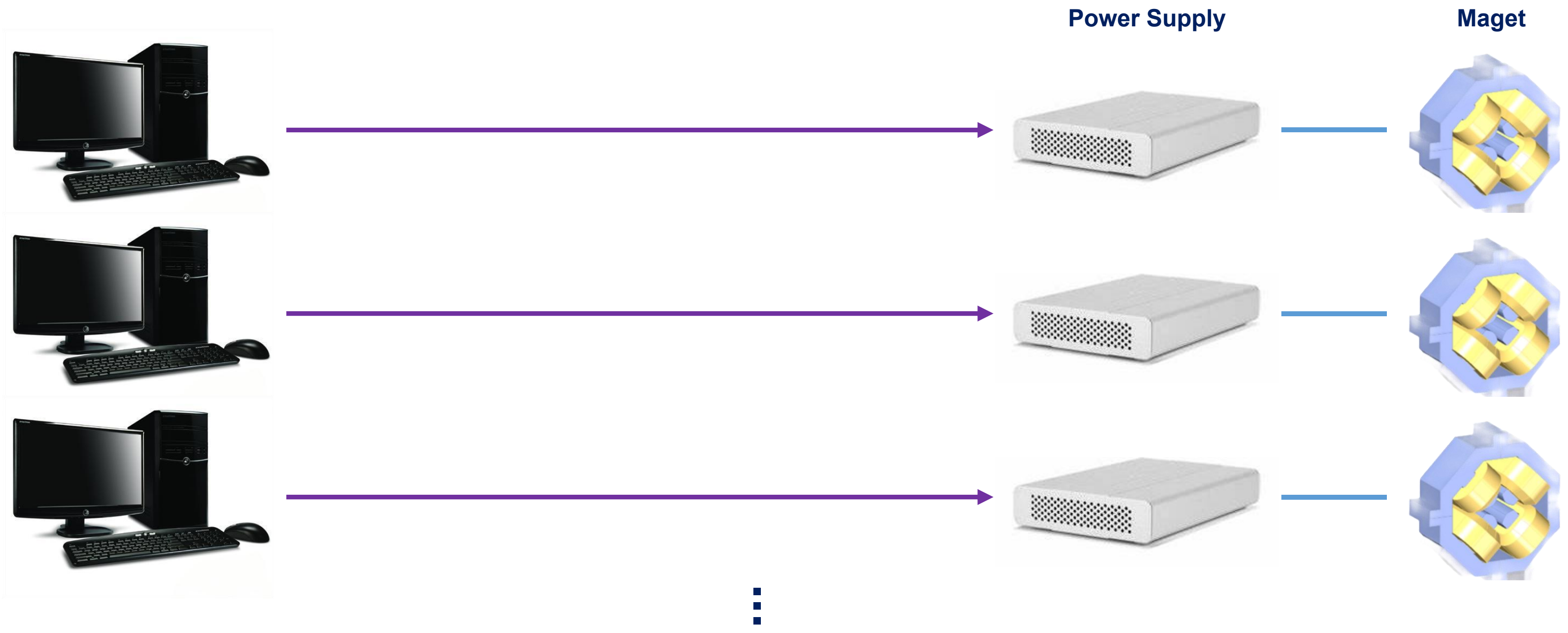
# What is an accelerator control system?



# What is an accelerator control system?


If there is **1** power supply, we only need to consider the simple **connection** and usable **software**.

# What is an accelerator control system?




If there are **1,000** or **10,000** devices, we need to consider the control system **architecture**, communication **network**, hardware **platform**, software **framework**, user-friendly interface, historical data storage & analysis tools, fault alarm and diagnosis, interlock or timing system. It is also necessary to ensure the large-scale system's **scalability**, **maintainability** and **reliability** .

# Control System Architecture

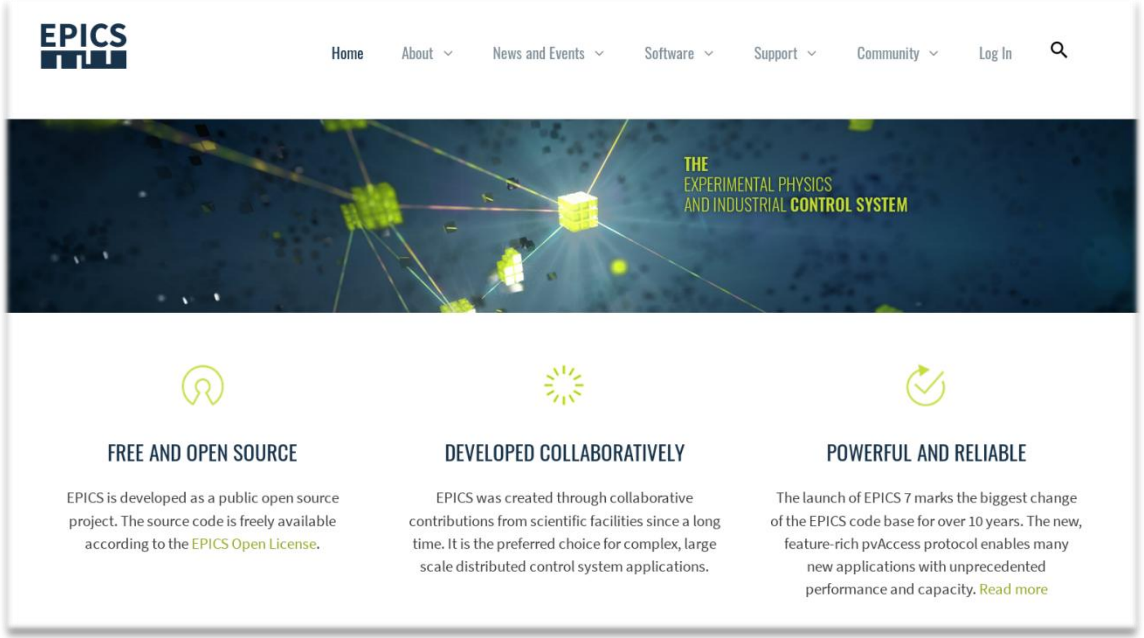
| | |
|---|---|
| Electrical cabinets | ~1,000 |
| Electrical equipments | ~10,000 |
| Hardware signals | ~100,000 |
| Software signals | ~1,000,000 |

**1**

- The **architectures**, **protocols** and **standards** are crucial for the design and implementation of large-scale control system **integration**.
- Adopt **mature technologies** and **standardized products** to minimize the R&D costs and timelines, while ensuring system stability and reliability.
- EPICS, TANGO, DOOCS, TINE, MADOCA, as well as commercial SCADA systems.

# EPICS

- Experimental Physics and Industrial Control System: Control System **Architecture** + Software **Toolkit** + **Collaboration**
- EPICS is a set of tools and applications which provide a **infrastructure** for use in building distributed control systems.
- EPICS uses **Client/Server** and **Publish/Subscribe** techniques to communicate between the various computers.
- EPICS is developed through a **collaborative** open-source process where anyone is free to contribute to the **community**.
- EPICS is provided under an **open source** license called the EPICS Open License, which is similar to the BSD license.



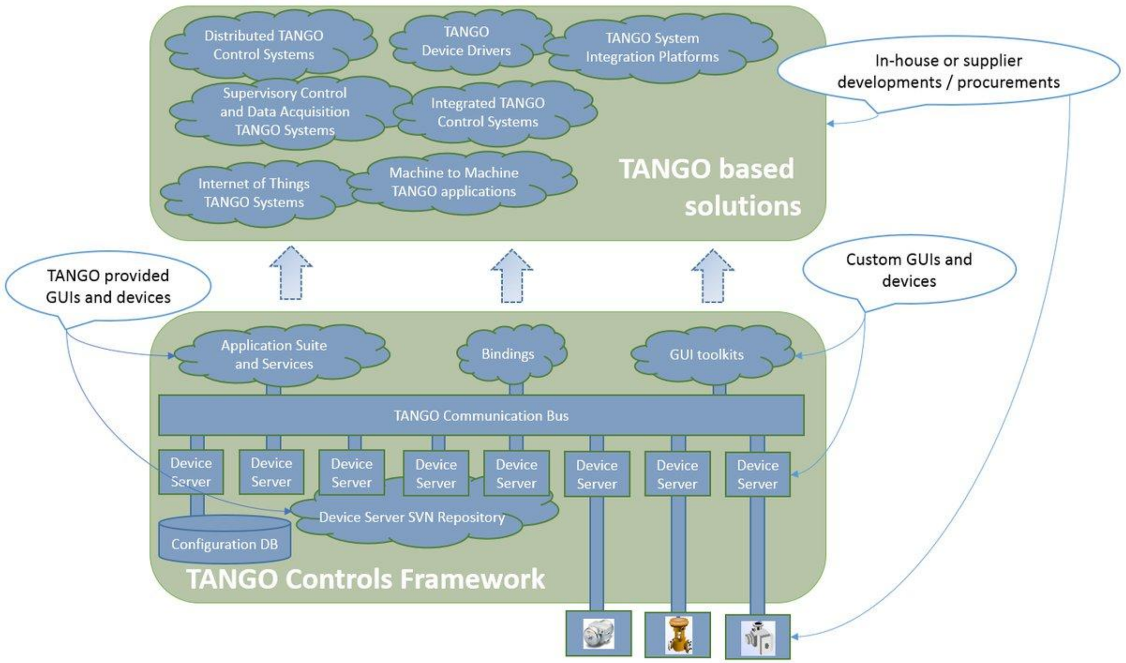

- BEPC, KEKB, RHIC, FRIB
- ESRF, DLS, SSRF, HEPS, HALF
- LCLS-I/II, SwissFEL, SHINE
- SNS, ESS, J-PARC, CSNS
- LIGO, ITER, EAO, ASKAP
- most facilities in Asia

https://epics.anl.gov/

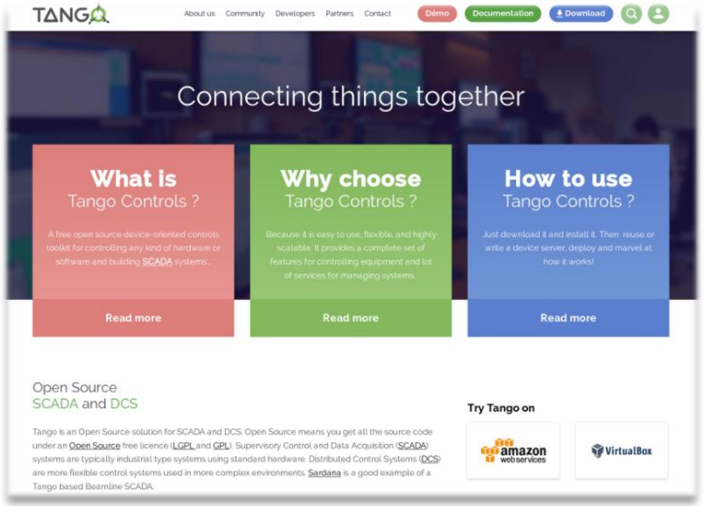
https://epics-controls.org/

# Tango Controls

- Tango Controls is a **toolkit** for connecting hardware and software together.
- It is a mature software which is used by tens of sites to run highly complicated accelerator complexes and experiments 24 hours a day.
- It provides full support for C++, Python and Java.
- It is **free** and **open source**. It is ideal for small and large installations.




ESRF, SOLEIL
ELETTRA, ALBA
PETRA III, MAX-IV
ELI, CEA, SKA
...

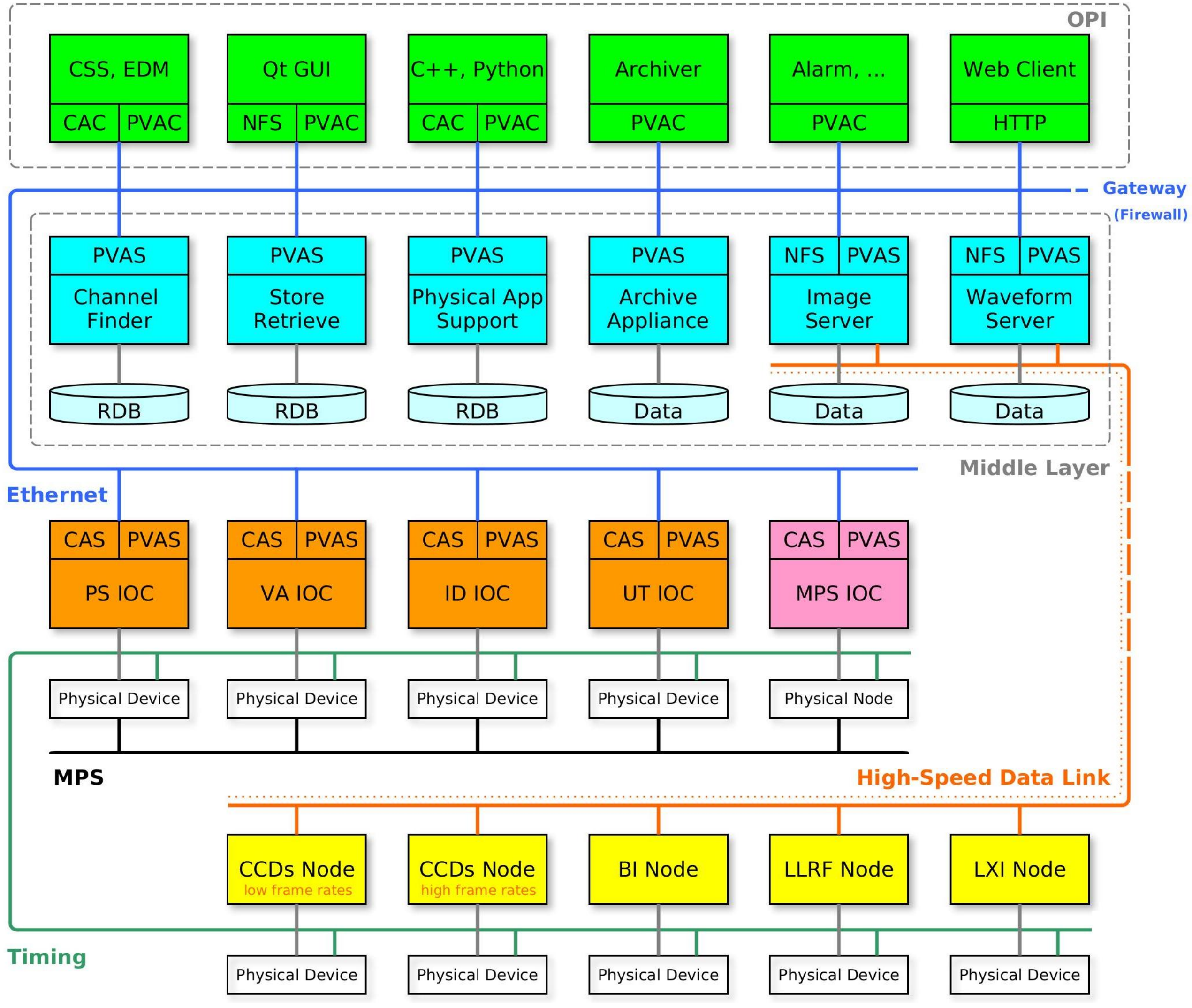



| | | | | |
|---|---|---|---|---|
| **150+** active members | **500+** device classes | **3 Million** lines of code | **1 000+** downloads of the core | **40+** international partners |

https://www.tango-controls.org/

# SHINE Accelerator Control System

- ❑ Network
- ❑ Control Platform
- ❑ Device Control
- ❑ Data Acquisition
- ❑ Machine Protection
- ❑ Timing System
- ❑ Feedback System
- ❑ Data Storage
- ❑ User Interface
- ❑ High Level Software
- ❑ Control Rooms
- ❑ …




Open source large-scale distributed control system, EPICS V7 + White Rabbit

EPICS

SANGFOR

Deepin

python

HDF5

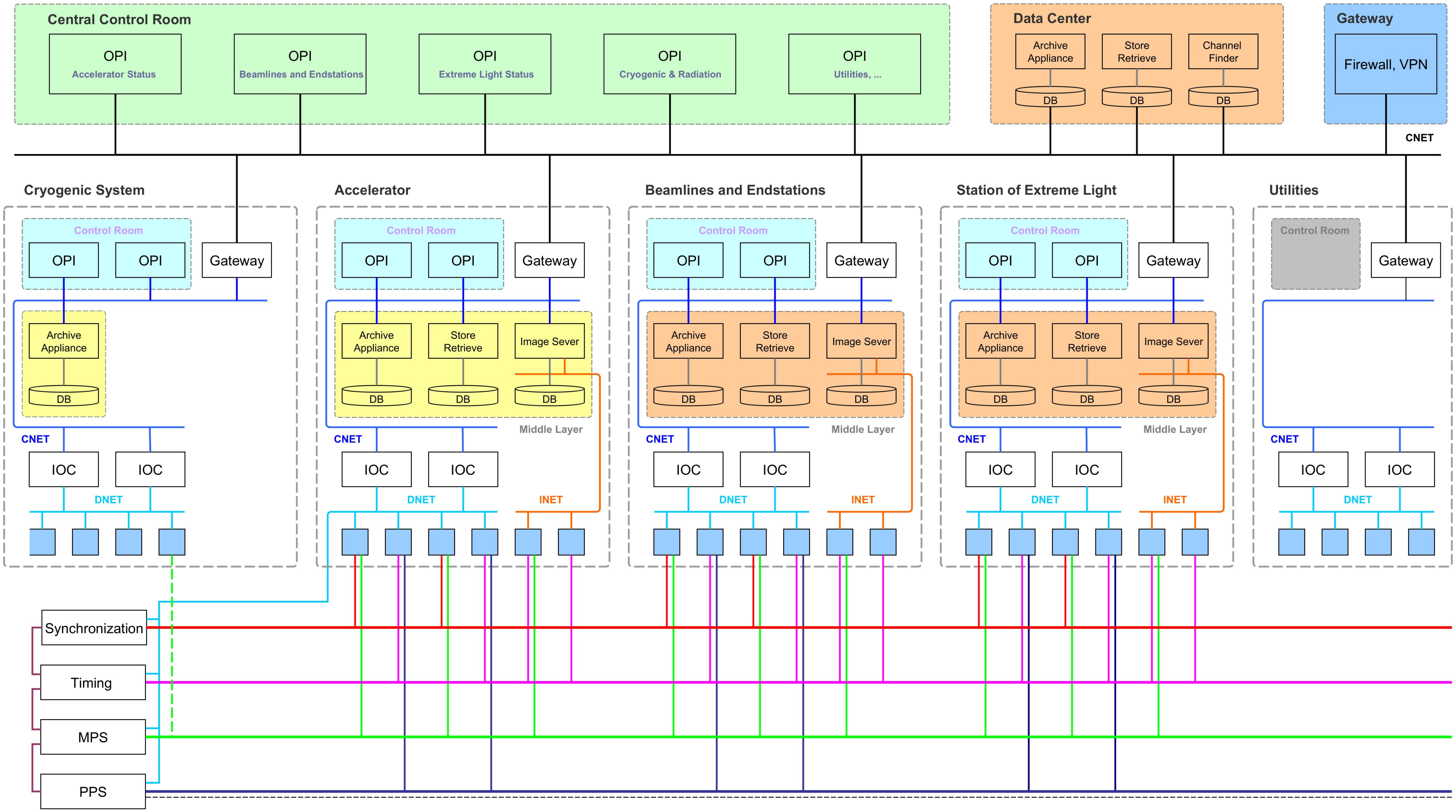
Grafana

GitLab

kafka

mongoDB

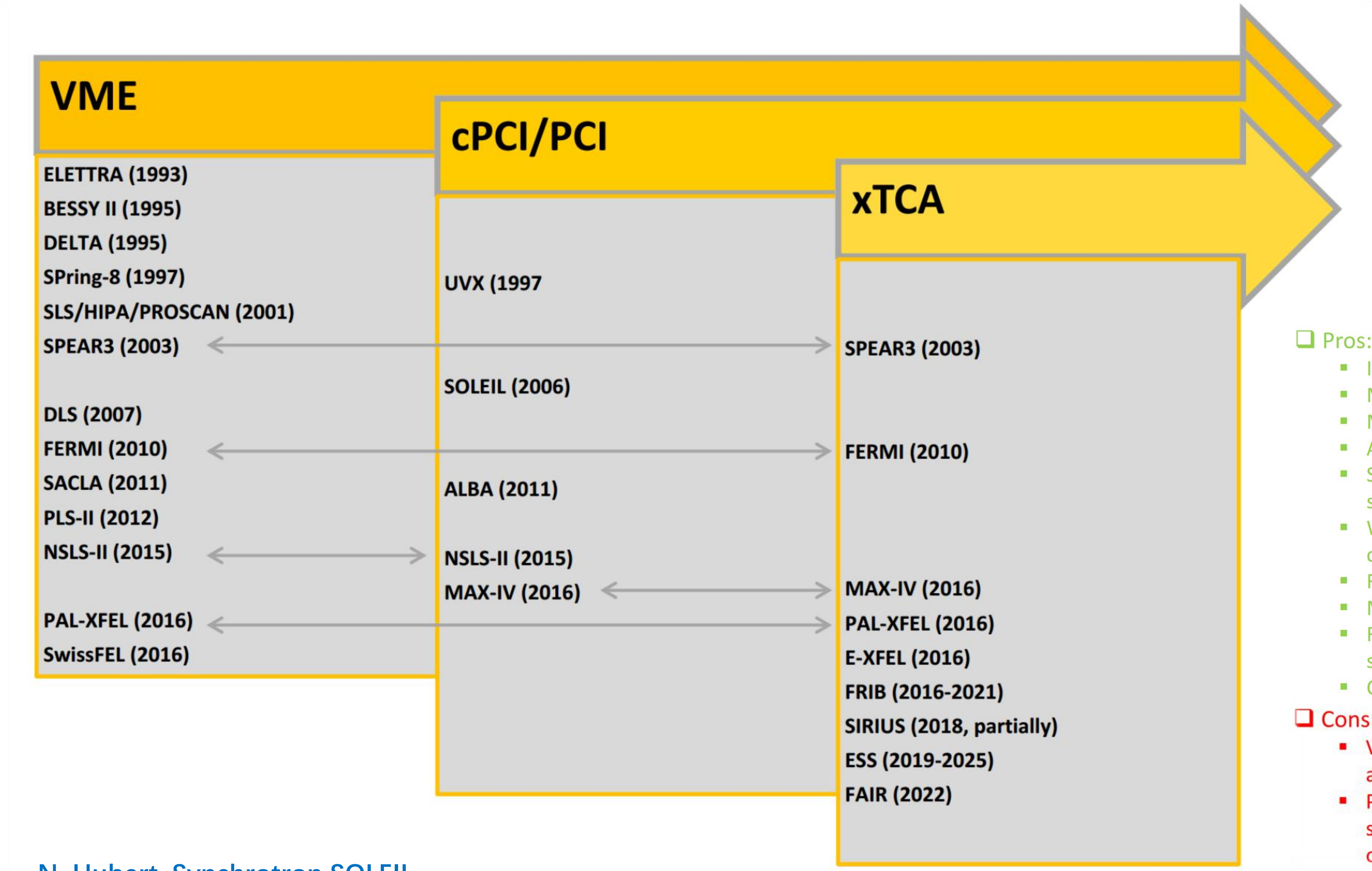
# SHINE Control System



# Hardware Platform

# Hardware Platform

NIM

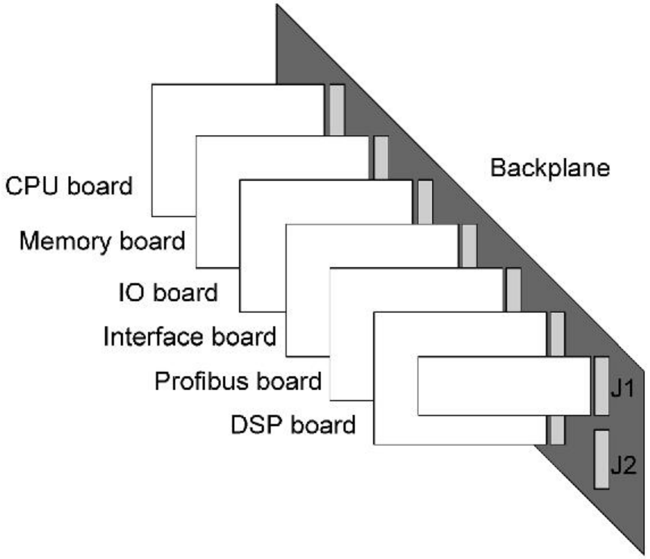


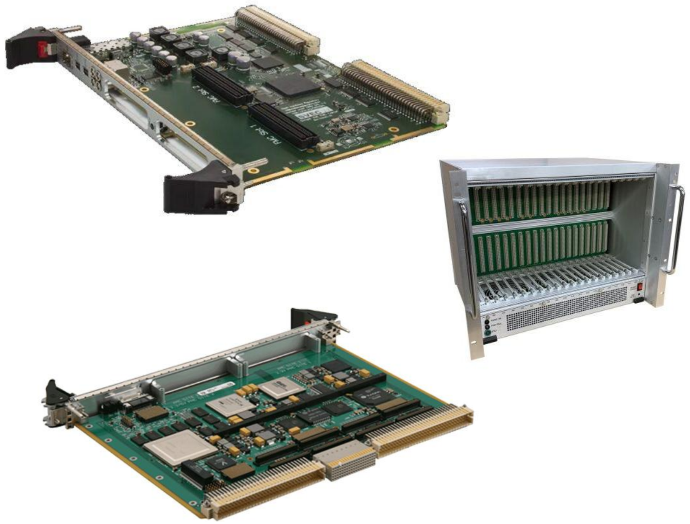

N. Hubert, Synchrotron SOLEIL

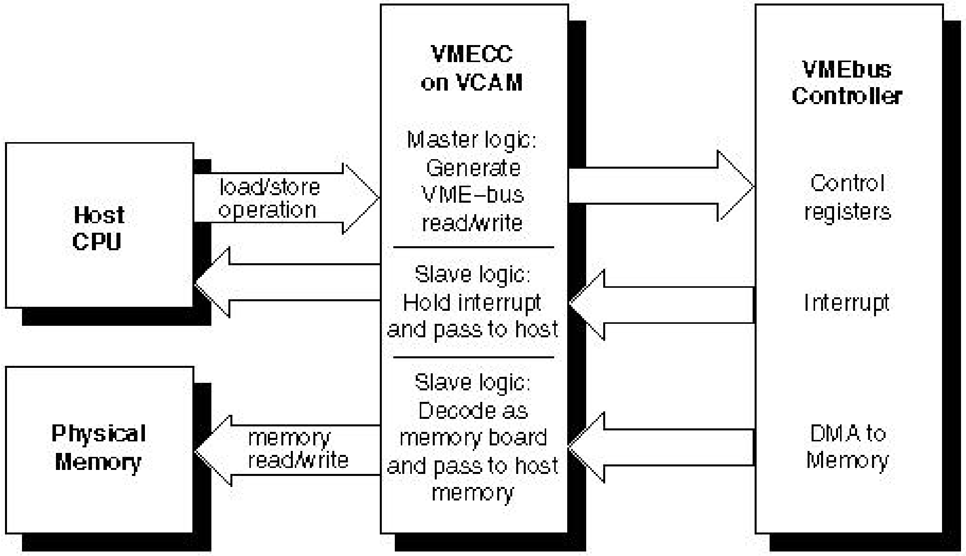
- Pros:
  - Integration
  - Maintenance and long term support
  - Management and control
  - Availability from industry
  - Standard boards: CPU, ADC, power supplies.
  - Wider user community (possible collaborations)
  - Reliability (if mature): long MTBF
  - Modularity: small MTTR
  - Redundancy (if supported by the standard)
  - Cost
- Cons:
  - Volume and cost for small application
  - Performances (bandwidth, bus speed sometimes limited for older standards)

# VME

- VME (**VERSA Module Eurocard**) initially developed starting in 1981, provides a **backplane-based** architecture for integrating processors and I/O modules, and instrumentation.
- High Reliability – Designed for industrial, aerospace, and defense use.
- Modular Design – Supports **6U** (160mm × 233mm) and **3U** (100mm × 160mm) Eurocard modules.
- **Parallel Bus Architecture** – 32-bit data/address bus (**VME32**) or 64-bit (**VME64x**).
- Standardized – IEEE 1014 (VME32), ANSI/VITA 1 (VME64x).

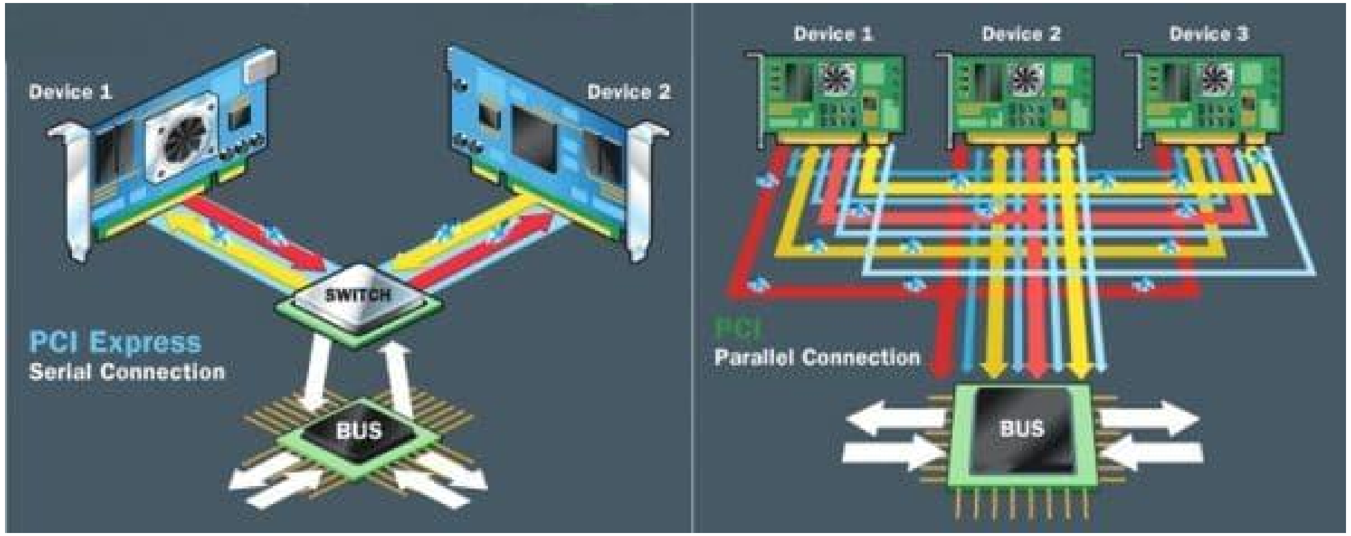


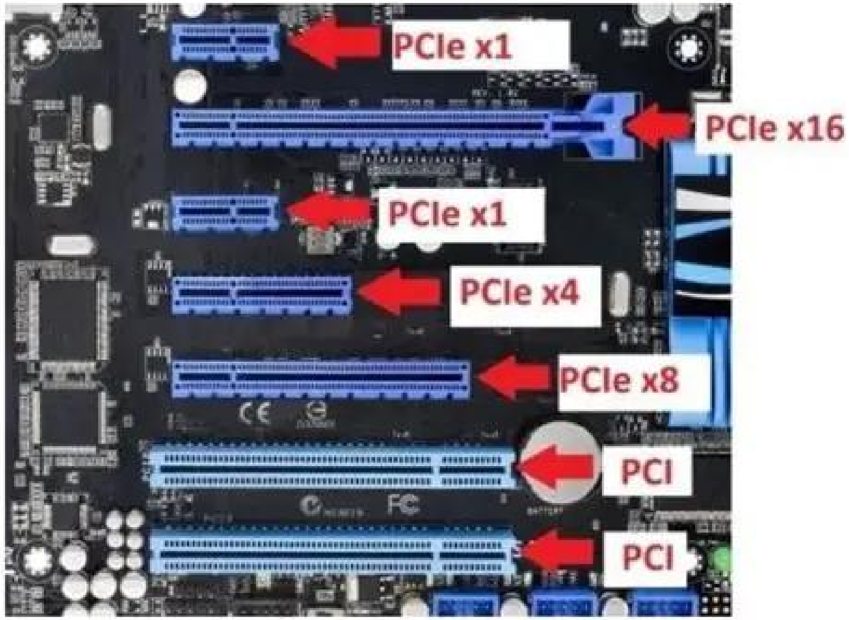





# PCIe

- PCIe (**Peripheral Component Interconnect Express**) is a high-speed **serial bus standard** used to connect hardware devices (GPUs, SSDs, NICs) to a computer's motherboard.
- It replaced older **PCI** buses, offering higher bandwidth, lower latency, and scalability.
- High Speed – From 250 MB/s (Gen1 x1) to ~64 GB/s (Gen5 x16).
- **Serial** Point-to-Point – No bus contention (unlike **parallel PCI**).
- Scalable Lanes – x1, x4, x8, x16 configurations.

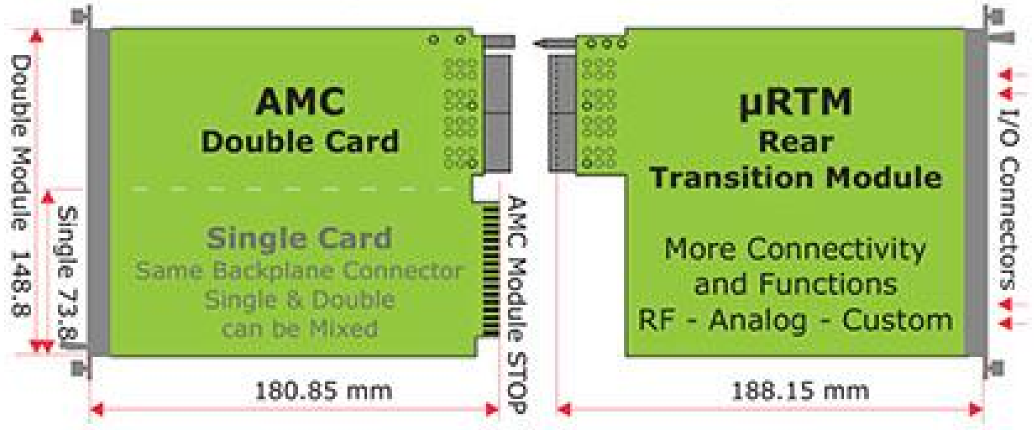



https://pcguide101.com/motherboard/what-are-pcie-slots/

# ATCA

- ATCA (**Advanced Telecommunications Computing Architecture**), also known as PICMG 3.x, is an open standard for high-performance, modular telecom and datacenter hardware.
- Carrier Card – FPGA, memory, backplane connections
- AMC (**Advanced Mezzanine Card**) – ADCs, DACs, high performance front end electronics
- RTM (**Rear Transition Module**) – General purpose IO,extra networks,miscellaneous





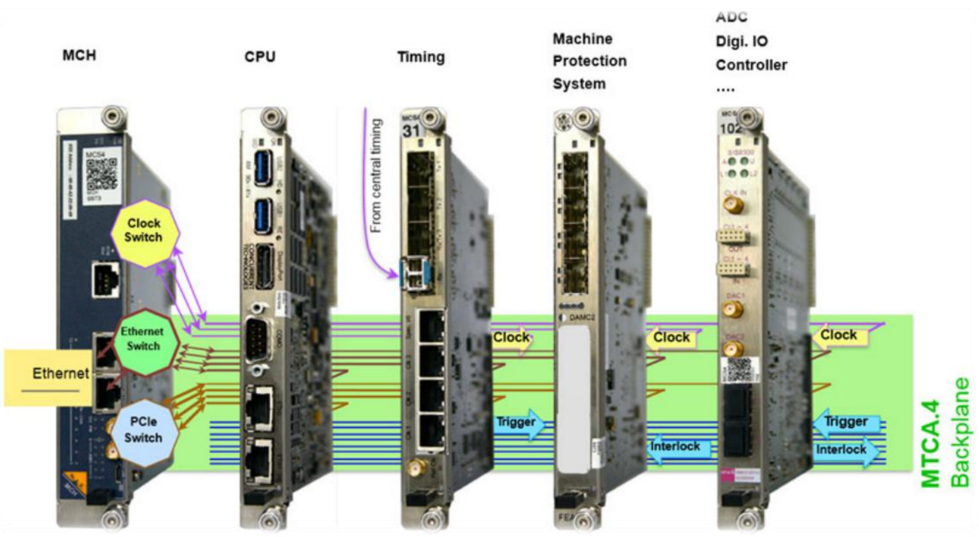
http://www.atcaworld.com/amc-overview.html

# MicroTCA

- **MicroTCA (µTCA)** is a compact, modular version of the AdvancedTCA (ATCA) standard.
- It leverages AMCs as its core building blocks while eliminating the need for a full ATCA chassis.
- Cost-Effective – **Lower price** point than ATCA while maintaining reliability.
- Compact & Scalable – **Smaller** than ATCA (1U to 6U rack units).
- MicroTCA/ATCA Workshop for Large Scientific Facility Control (Sep 2025)

µTCA®

| Feature | MicroTCA (µTCA) | ATCA |
| --- | --- | --- |
| Form Factor | 1U-6U rack | 8U-14U rack |
| Core Block | AMC (Advanced Mezzanine Card) | Full-size ATCA Blades |
| Speed | PCIe Gen3/4, 10G/40G Ethernet | PCIe Gen4, 100G+ Ethernet |
| Target Market | Telecom edge, industrial | Telecom core, data centers |
| Cost | Medium | High |

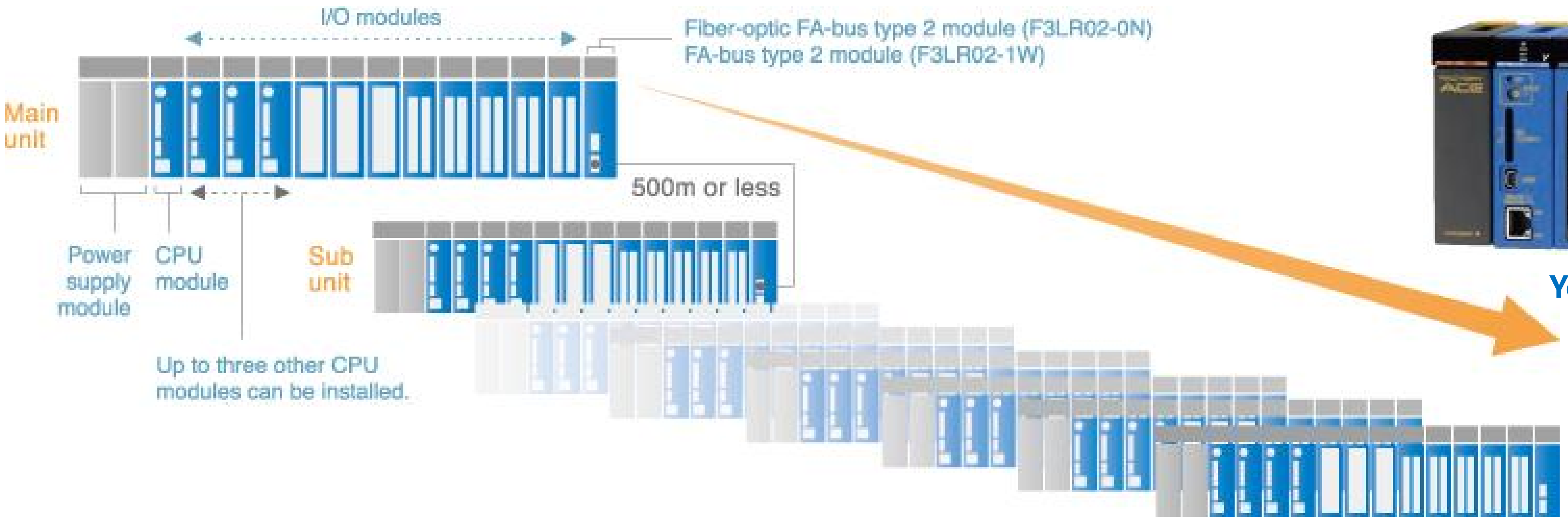


# Hardware Bus

| Feature | NIM | VME | PCIe | cPCI | PXI | ATCA | µTCA |
|---|---|---|---|---|---|---|---|
| Year Introduced | 1960s | 1981 | 2003 | 1994 | 1997 | 2002 | 2006 |
| Primary Use | Nuclear Physics instruments | Military Telecom | Consumer Enterprise | Industrial Automation | Test Measurement | Telecom & Data Centers | Telecom Edge & Industrial |
| Bus Type | Analog + Digital Signals | Parallel (VME64x) | Serial (PCIe) | Parallel (PCI) | PCI/PCIe + Triggers | Serial (Ethernet/PCIe) | AMC-based (PCIe/Ethernet) |
| Speed | kHz-MHz | 40 MB/s (VME32) | Up to 64 GB/s (PCIe Gen5) | 133 MB/s (32-bit PCI) | 24 GB/s (PCIe Gen3 x16) | 100G+ (Fabric) | 40G+ (PCIe Gen3) |
| Form Factor | Modular Bin | 3U/6U Eurocard | Add-in Card | 3U/6U Eurocard | 3U/6U CompactPCI | 8U-14U Chassis | 1U-6U Chassis |
| Status | Legacy (niche use) | Replaced by VPX | Dominant in PCs/servers | Replaced by cPCI Serial | Active | Active | Active |

# PLC

- **Programmable Logic Controller** is an industrial computer with inputs and outputs used in the control and automation of industrial processes.
- Dual Redundancy – Supports hot-swappable CPUs & I/O modules for fault tolerance.
- Advanced Communication – Protocols like **Ethernet/IP**, **Modbus**, **PROFIBUS**, **OPC UA**.

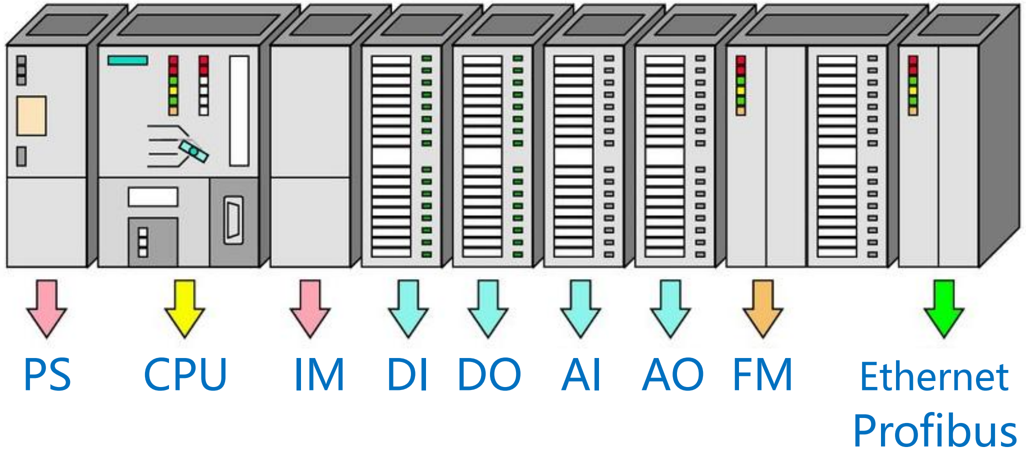



Yokogawa PLC

https://www.yokogawa.com/solutions/products-and-services

# Siemens PLC

- **Siemens S7** refers to a family of programmable logic controllers (PLCs) and automation systems developed by Siemens AG for industrial control applications.
- Modular Design – Expandable I/O (digital, analog, specialty modules).
- Communication Protocols – PROFINET, Ethernet/IP, Modbus, OPC UA.
- **Motion Control** – Supports servo drives.

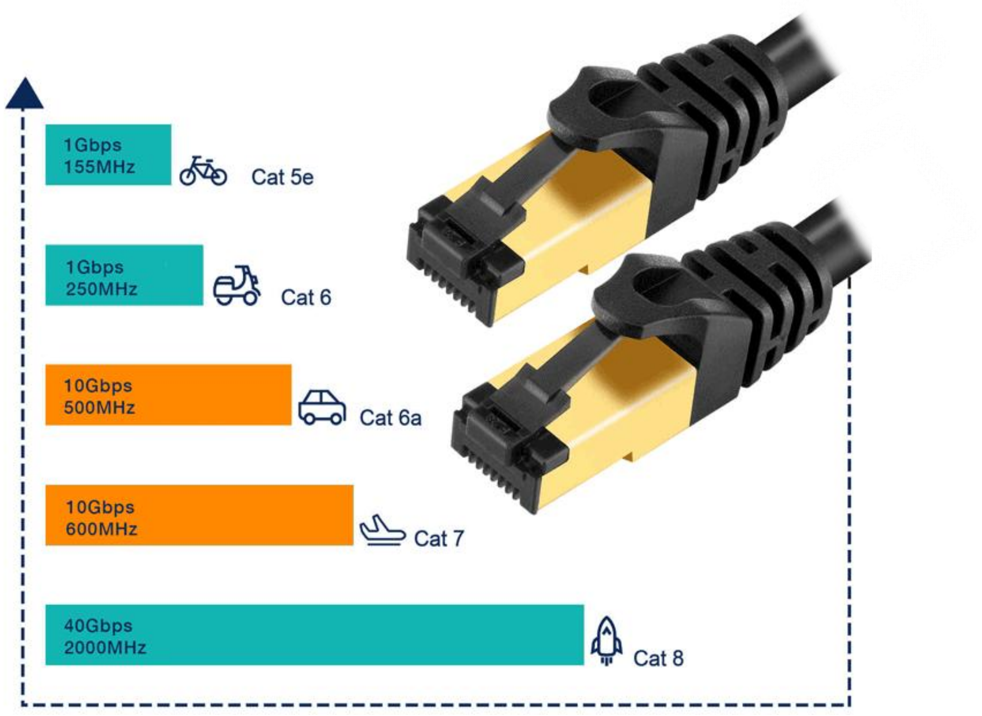



SIEMENS

# Interface & Protocol

# Ethernet

- Ethernet is a widely used wired networking technology that connects devices in a **Local Area Network (LAN)** or Wide Area Network (WAN).
- Physical Connection
  - Uses twisted-pair cables or fiber optics
  - Connects via **RJ45** ports or **SFP** ports
- Speed & Performance
  - Ranges from 10 Mbps to 400 Gbps (latest standards)
  - Common speeds: **1 Gbps**, **10 Gbps**, **25 Gbps**
- Topology
  - Traditionally used a bus topology (old coaxial cables)
  - Modern Ethernet uses **star topology** (central switch)
  - **Ring topology** is used by some small facilities




https://storage.googleapis.com/dufgkmmlknqvre/what-type-of-ethernet-cable-do-i-need-for-poe.html

# Serial

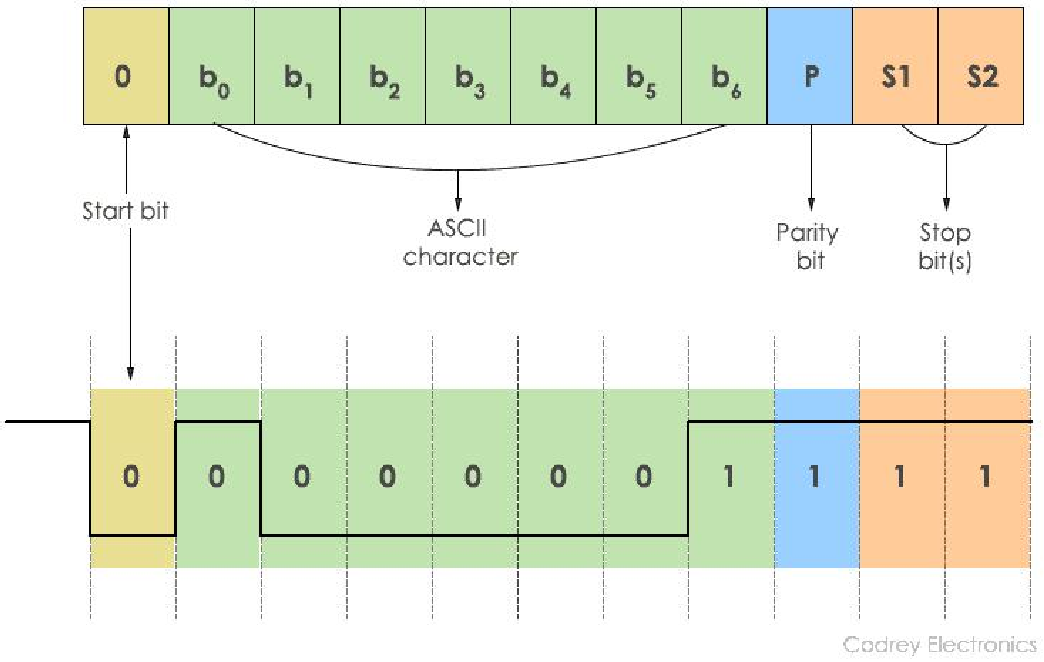
- RS232 (**Recommended Standard**) is an old **serial communication protocol** developed by EIA (Electronics Industry Alliance) / TIA (Telecommunications Industry Association) in 1962.
- RS232 describes the common voltage levels, electrical standards, operation mode and number of bits to be transferred from transmitter to receiver.


| Pin | Signal Direction | Signal Name | Signal Function |
|---|---|---|---|
| 1 | ← | CD | Carrier Detect |
| 2 | ← | RxD | Receive Data |
| 3 | → | TxD | Transmit Data |
| 4 | → | DTR | Data Terminal Ready |
| 5 | — | GND | Ground |
| 6 | ← | DSR | Data Set Ready |
| 7 | → | RTS | Request To Send |
| 8 | ← | CTS | Clear To Send |
| 9 | ← | RI | Ring Indicator |

→ Transmitted from DTE Device
← Received by DTE Device

| Pin | Signal Direction | Signal Name | Signal Function |
|---|---|---|---|
| 1 | → | CD | Carrier Detect |
| 2 | → | TxD | Transmit Data |
| 3 | ← | RxD | Receive Data |
| 4 | ← | DTR | Data Terminal Ready |
| 5 | — | GND | Ground |
| 6 | → | DSR | Data Set Ready |
| 7 | ← | CTS | Clear To Send |
| 8 | → | RTS | Request To Send |
| 9 | → | RI | Ring Indicator |

→ Transmitted from DCE Device
← Received by DCE Device

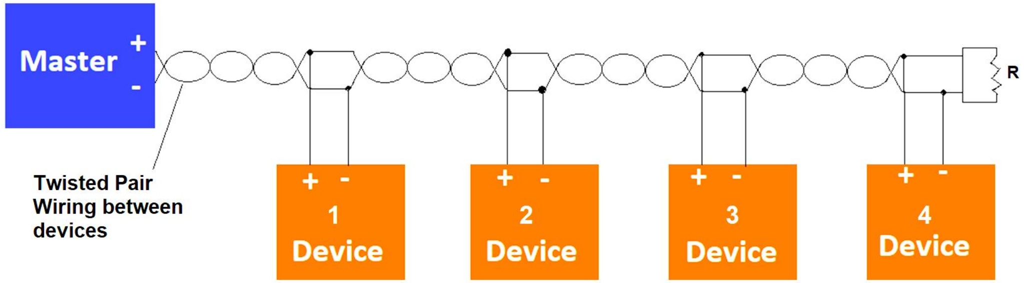



https://www.codrey.com/embedded-systems/rs232-serial-communication/

# Serial

- RS422 is designed for **faster** data transmission over **longer** distances compared to RS232.
- RS485 can be considered an upgraded version of RS422, allowing one device to control up to 32 devices. It became the **standard physical layer** for automation protocols such as **Modbus RTU** and **Profibus**.





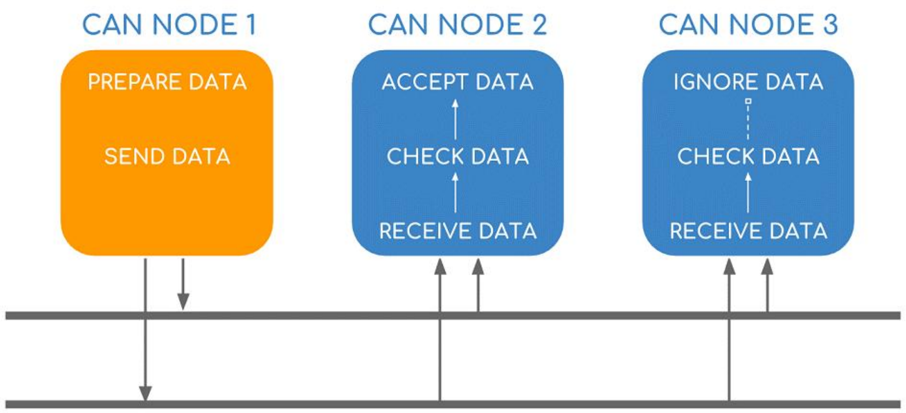

RS 422 / RS 485 Multi Drop Network Connection

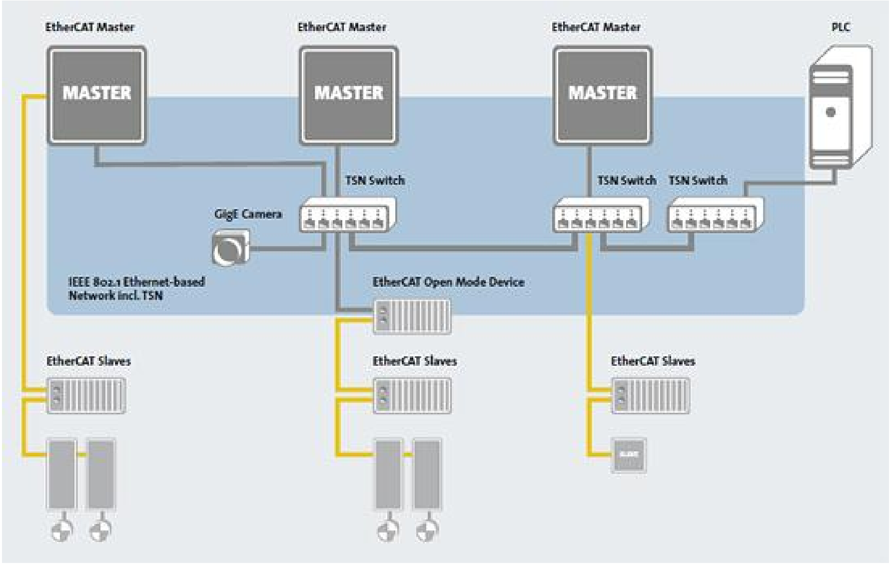
| Port name | RS-232 | RS-422 | RS-485 |
|---|---|---|---|
| Transfer type | Full duplex | Full duplex | Half duplex (2 wires), full duplex (4 wires) |
| Maximum distance | 15 meters at 9600 bps | 1200 meters at 9600 bps | 1200 meters at 9600 bps |
| Contacts in use | TxD, RxD, RTS, CTS, DTR, DSR, DCD, GND* | TxA, TxB, RxA, RxB, GND | DataA, DataB, GND |
| Topology | Point-to-Point | Point-to-Point | Multi-point |
| Max. Number of connected devices | 1 | 1 (10 devices in receive mode) | 32 (with repeaters larger, usually up to 256) |

https://ipc2u.com/articles/knowledge-base/the-main-differences-between-rs-232-rs-422-and-rs-485/

# Fieldbus

- Fieldbus is a a member of a family of **industrial digital communication networks** used to connect automation devices (sensors, actuators, PLCs, controllers).
- **PROFIBUS** (Process Field Bus)
  - Developed by: Siemens
  - Speed: Up to 12 Mbps (PROFIBUS DP)
- **CAN** Bus (Controller Area Network)
  - Robust, low-cost, peer-to-peer communication
- **EtherCAT** (Ethernet for Control Automation Technology)
  - Ultra-Fast Industrial Ethernet
  - Speed: Up to 100 Mbps with real-time performance
- …

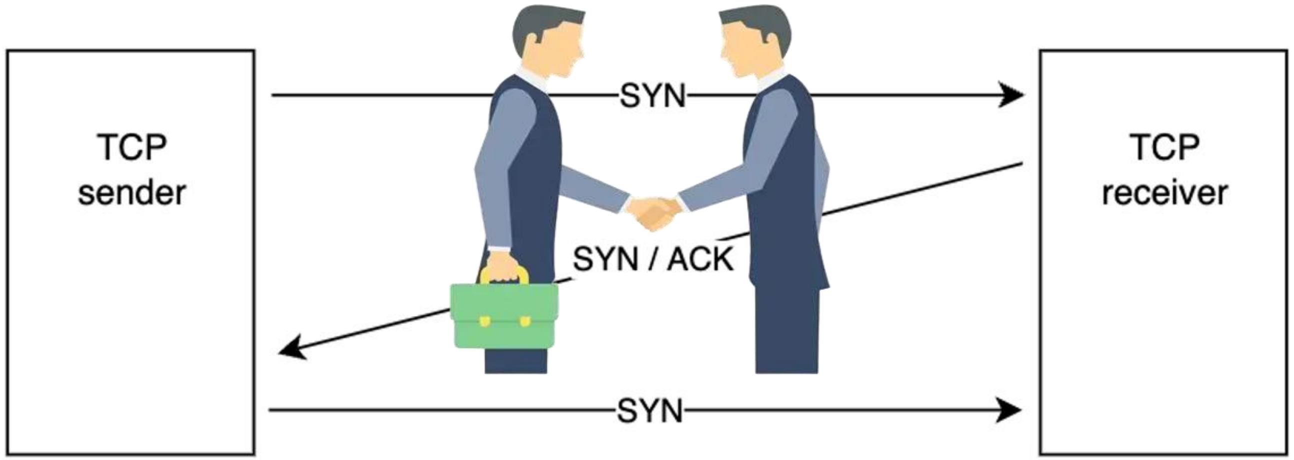


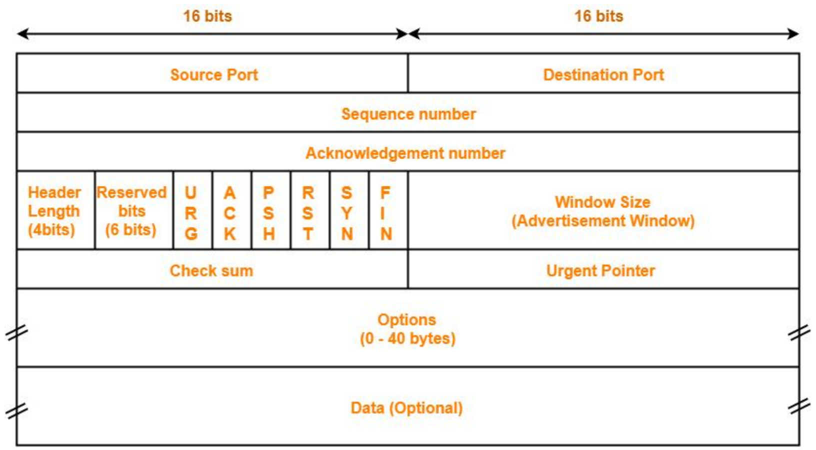

https://store.chipkin.com/articles/can-bus-protocol-10-minute-lesson
https://www.ethercat.org/en/ethercat_and_tsn.htm

# TCP

- **Transmission Control Protocol** is one of the core protocols of the Internet Protocol Suite (TCP/IP).
- Connection-Oriented – Establishes a **handshake** before data transfer.
- Reliable Delivery – Uses acknowledgments (ACKs) and retransmissions if packets are lost.
- Flow Control – Prevents sender from overwhelming receiver (sliding window mechanism).
- Error Detection – Uses checksums to detect corrupted data.
- Ordered Data Transfer – Ensures packets arrive in the correct sequence.



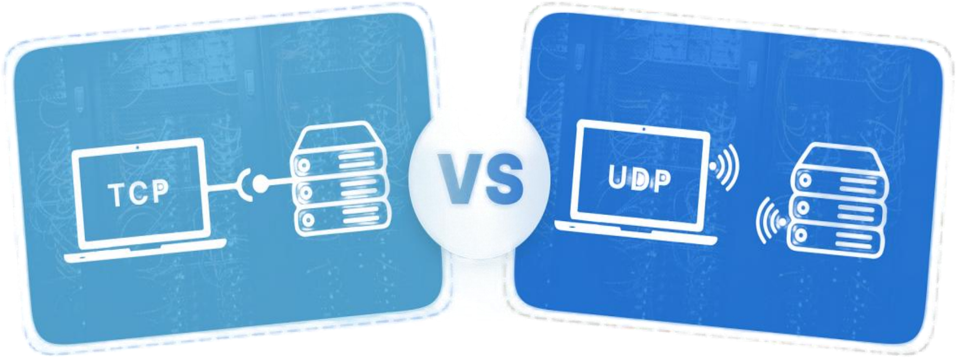

TCP 3 way handshake




TCP Header

# UDP

- **User Datagram Protocol** doesn't establish a connection before sending data, **faster** but less reliable.
- Connectionless – No handshake (SYN/ACK), just sends data.
- Low Latency – Minimal overhead, ideal for real-time apps.
- No Guaranteed Delivery – No retransmissions if packets are lost.
- No Ordering – Packets may arrive out of sequence.
- Supports **Broadcast / Multicast** – Can send data to multiple devices at once.




| Factor | TCP | UDP |
|---|---|---|
| Guarantees frame delivery | Yes | No |
| Acknowledgment used | Yes | No |
| Connectionless | No | Yes |
| Used when fast transmissions are required | No | Yes |
| Common TCP and UDP ports | DNS, HTTP, HTTPS, FTP, SMTP, Telnet | DNS, DHCP, TFTP, SNMP, RIP, VOIP |

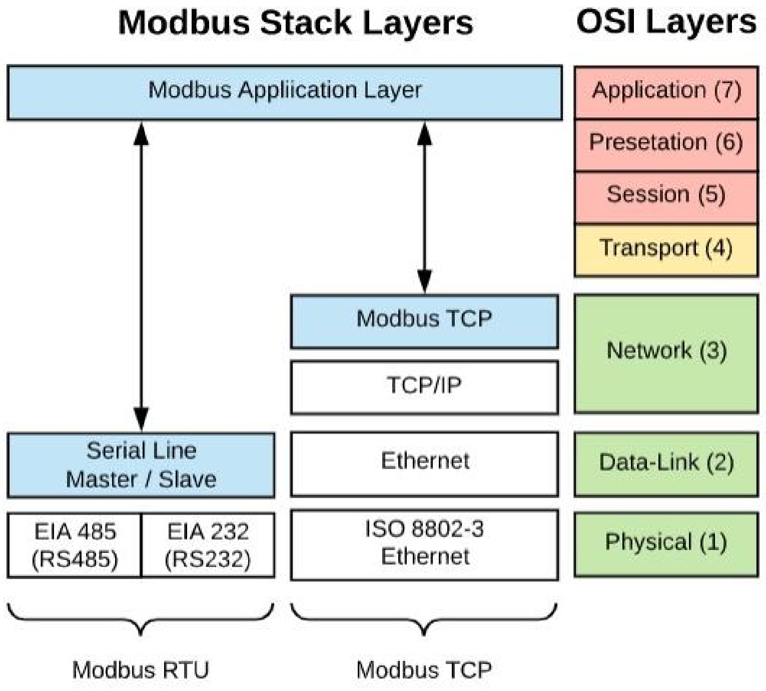
https://www.gumlet.com/learn/tcp-vs-udp/

https://www.netally.com/network-performance/tcp-vs-udp/

# Modbus

- Modbus is a **serial communication protocol** developed for industrial automation in 1979.
- It enables communication between electronic devices over **RS-232**, **RS-485**, or **Ethernet** (TCP/IP).
- Open Standard – Free to use, **widely** adopted in industrial systems.
- Master-Slave Architecture – One master device controls multiple slaves.
- Supports Multiple Variants – **Modbus RTU**, **ASCII**, **TCP**, ...
- Simple & Lightweight – Easy to implement.

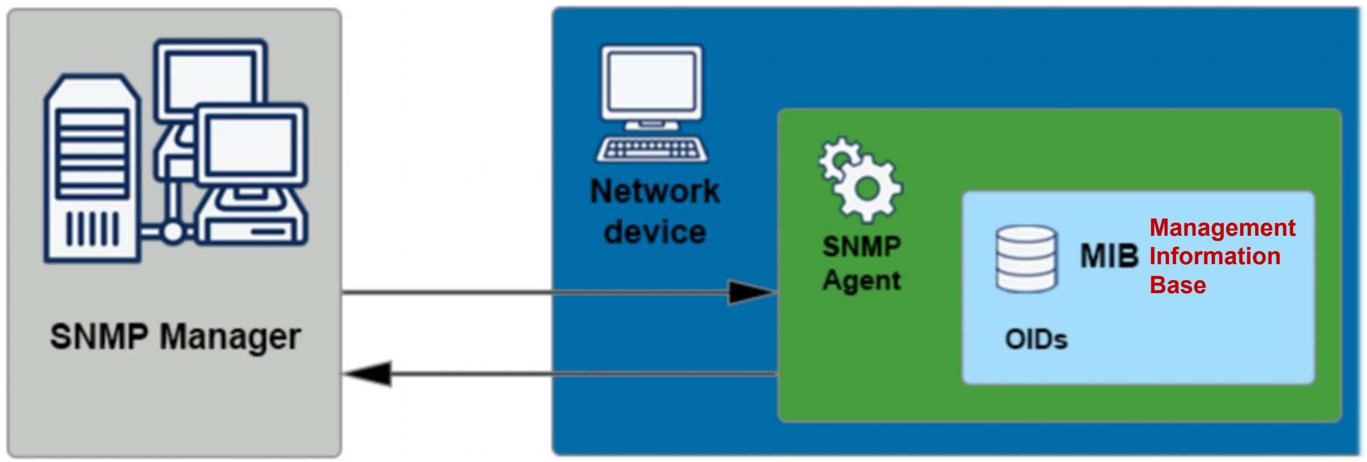
| Type | Description | Physical Layer |
|---|---|---|
| Modbus RTU | Binary encoding, compact & fast | RS-485 / RS-232 |
| Modbus ASCII | Human-readable (hexadecimal format). | RS-485 / RS-232 |
| Modbus TCP | Runs over Ethernet (TCP/IP) | Ethernet (RFC 1006) |
| Modbus Plus (MB+) | Proprietary high-speed version. | Token-passing network |

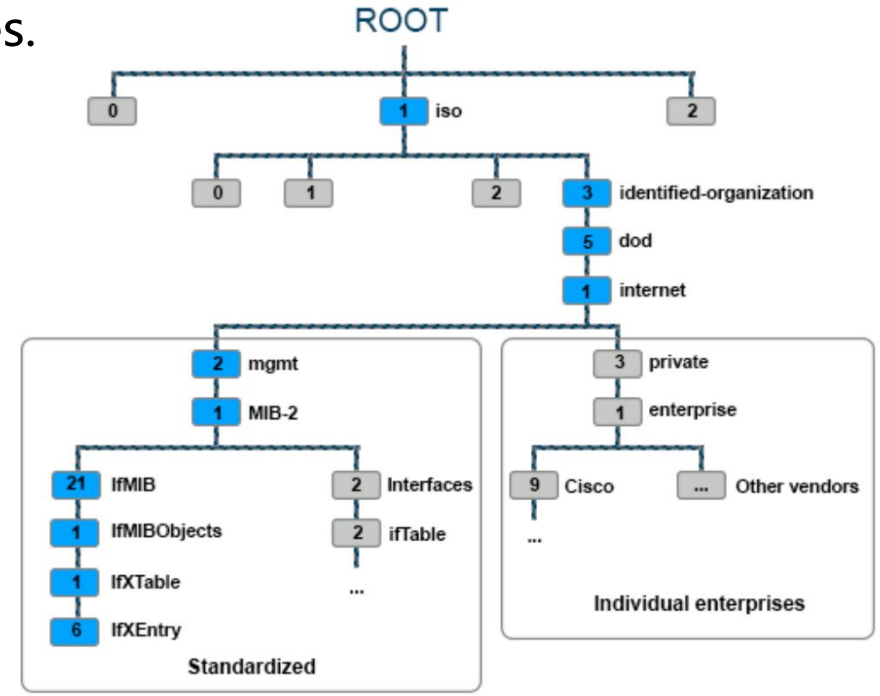



https://store.chipkin.com/articles/different-types-of-modbus-such-as-rtu-tcp-etc

# SNMP

- SNMP (**Simple Network Management Protocol**) is a standardized **protocol** used to monitor and manage **network devices** (switches, routers, servers, IoT devices, etc).
- Device Monitoring – Tracks bandwidth, CPU usage, memory, etc.
- Fault Detection – Alerts when issues occur (e.g., high temperature, link down).
- Remote Configuration – Changes settings on network devices.
- Scalable – Works in small LANs or large enterprise networks.

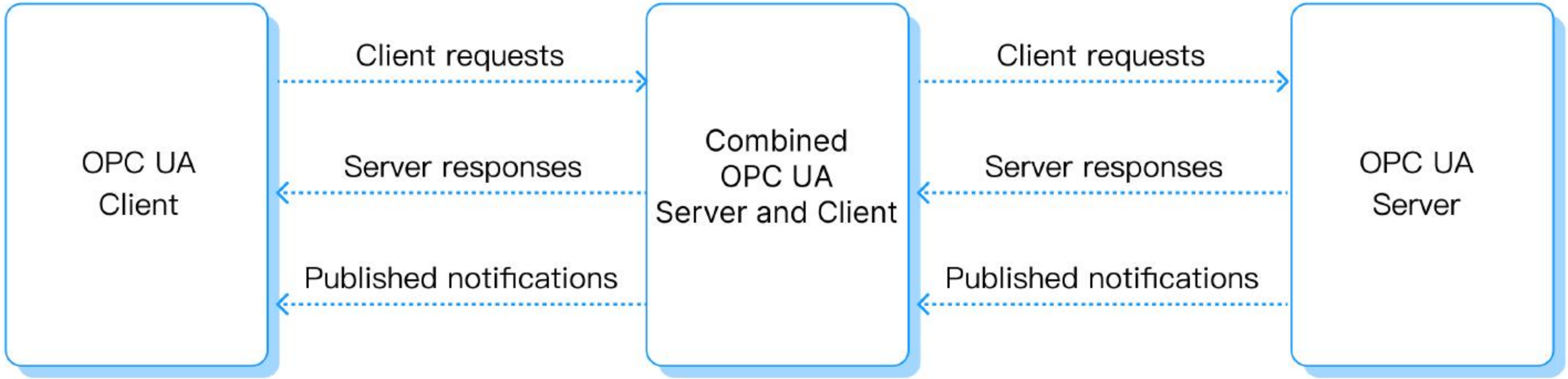



https://phoenixnap.com/kb/what-is-snmp

# OPC UA

- OPC UA (**Open Platform Communications Unified Architecture**) is a communication **protocol** designed for industrial automation.
- Unlike traditional OPC which relied on Windows COM/DCOM, OPC UA is **platform-independent**, secure, and scalable, making it ideal for **Industry 4.0** and **IoT** applications.
- Platform-Independent – Runs on Windows, Linux, embedded systems, and cloud.
- Transport Protocols – **OPC UA TCP**, **HTTPS/WebSockets**, **MQTT** (For lightweight IoT scenarios)

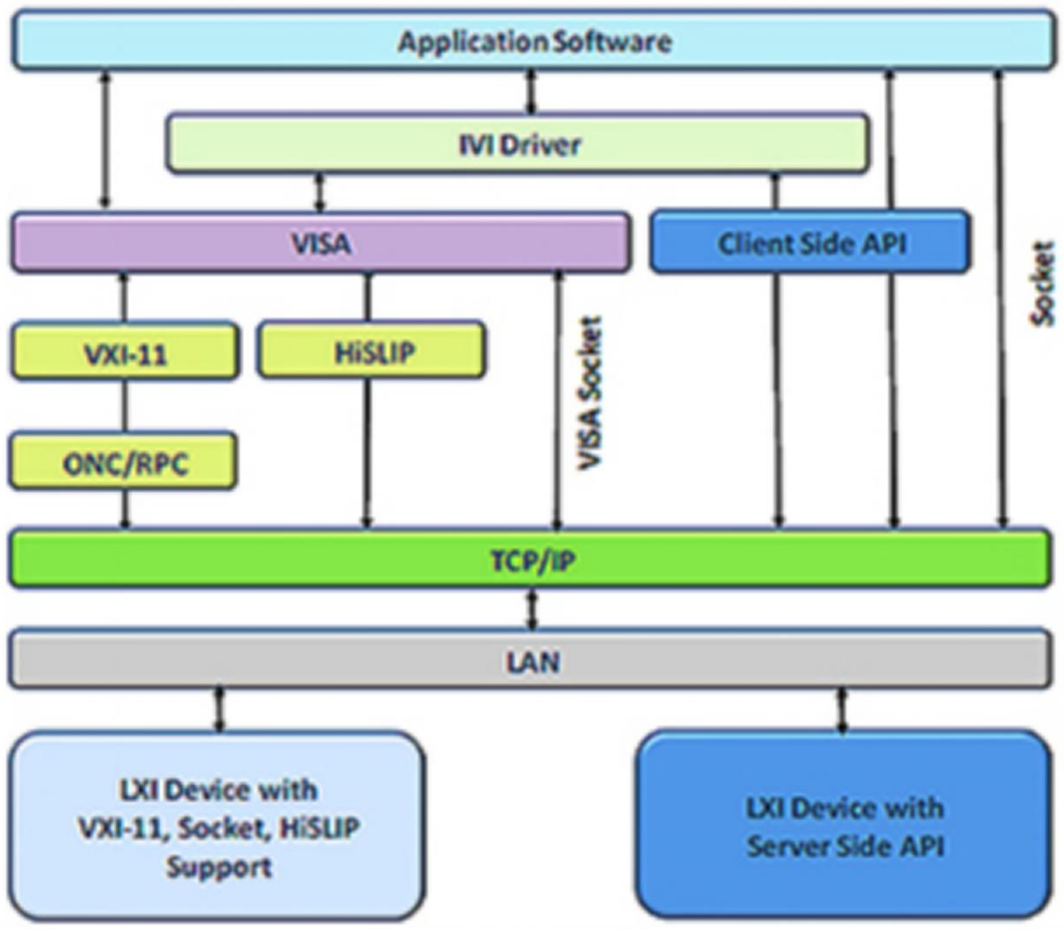



https://www.emqx.com/en/blog/opc-ua-protocol

# VXI-11

- VXI-11 is a standard **Ethernet-based** protocol for controlling test and **measurement instruments** (e.g., oscilloscopes, power supplies) over a LAN.
- It is part of the VXI (**VME eXtensions for Instrumentation**) standard and serves as an open, network-friendly alternative to GPIB (IEEE-488).

```python
import vxi11 # 导入vxi11包

def main():
    HOST = '192.168.0.100' # IP地址
    instr = vxi11.Instrument(HOST) # 连接仪器
    res = instr.ask('*idn?') # 发送询问指令
    print(res)

if __name__ == '__main__':
    main()
```

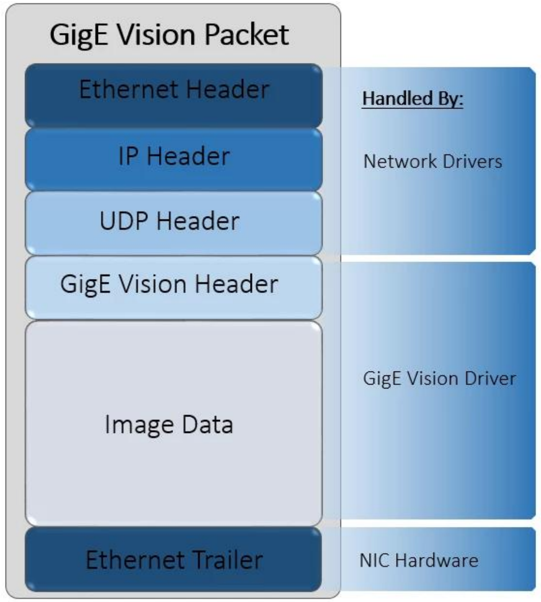


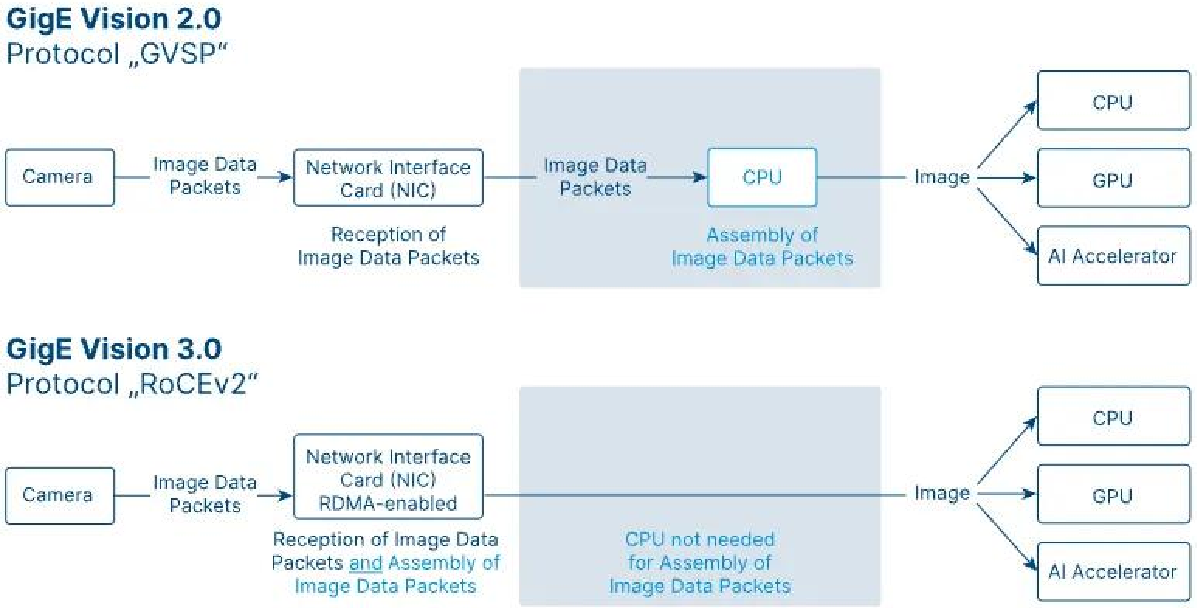

VISA：Virtual Instrument Software Architecture

https://lxistandard.org/

# GigE Vision

- GigE Vision is a standardized protocol for **high-speed camera** communication over Gigabit Ethernet (**GigE**), widely used in industrial imaging, machine vision, and robotics.
- High Speed – Uses Gigabit Ethernet (**1 Gbps**) or 10 GigE (**10 Gbps**) for fast image transfer.
- Standardized – Managed by the Automated Imaging Association (AIA).
- **PoE** Support – **Power over Ethernet** (IEEE 802.3af/at) for single-cable operation.



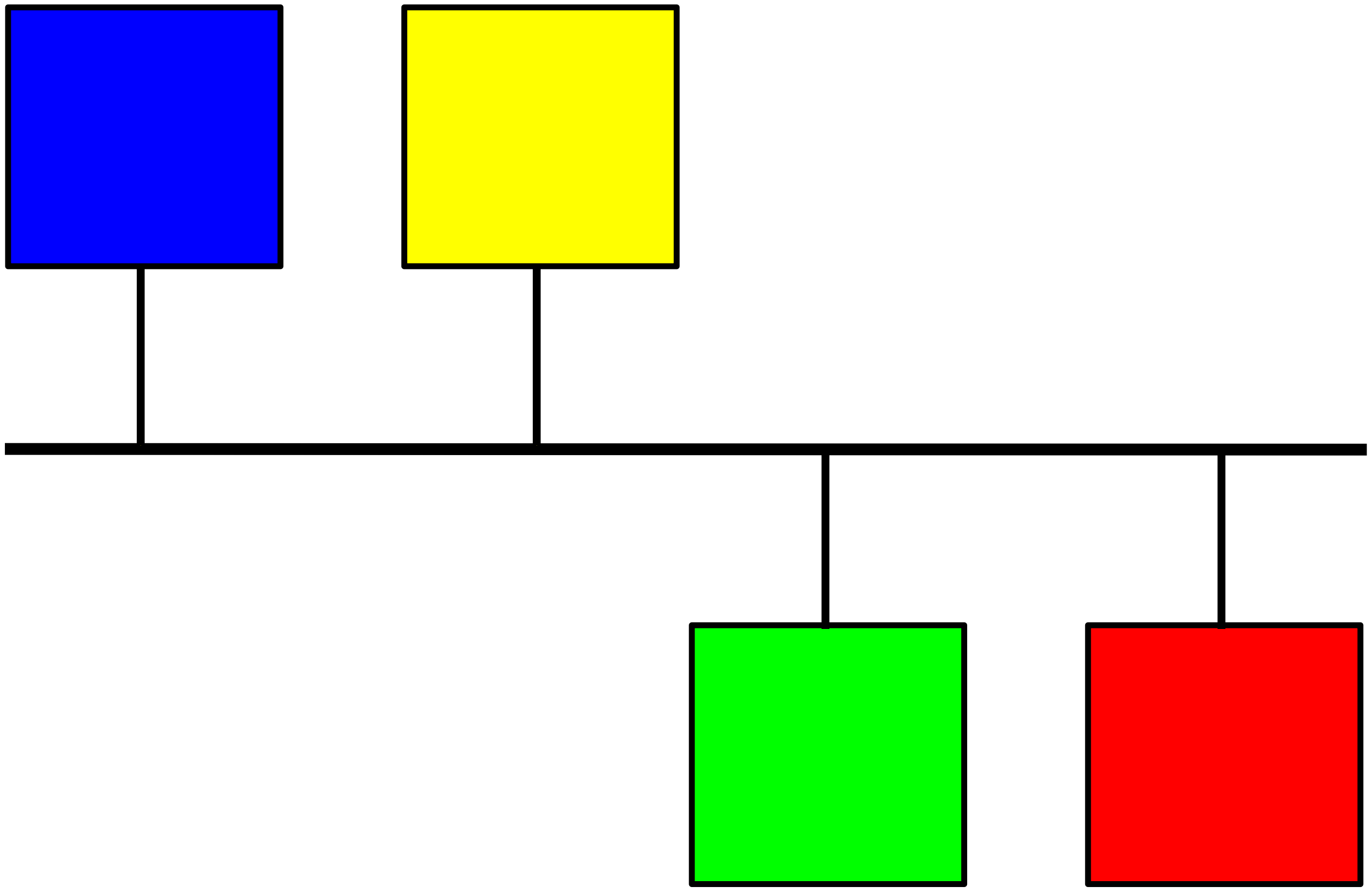

https://www.ni.com/

https://www.baslerweb.com/en/learning/gige-vision-3-0/

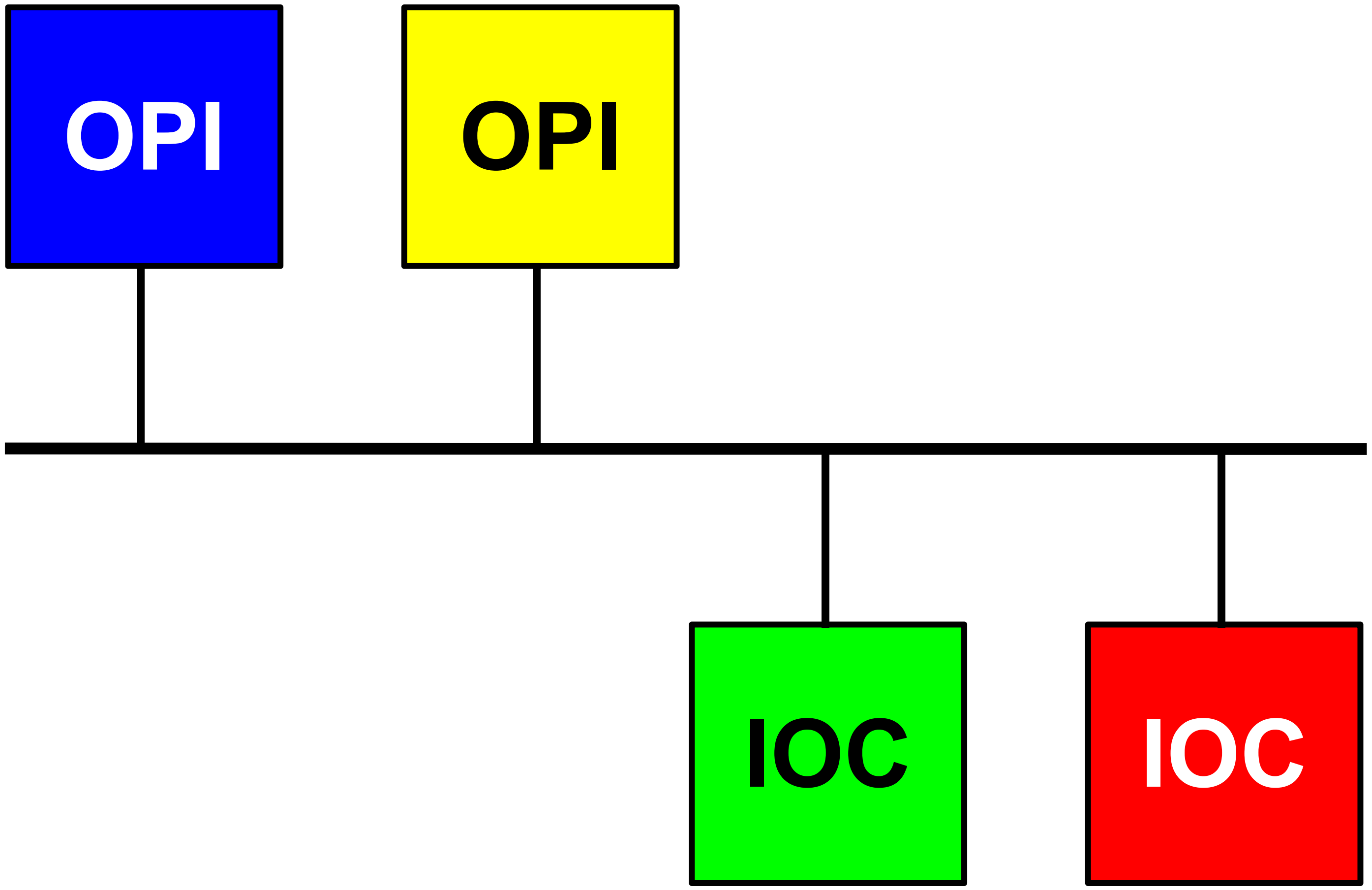
# EPICS Basic & Modules

Some materials and ideas are copied from Ken Evans (APS) and EPICS website

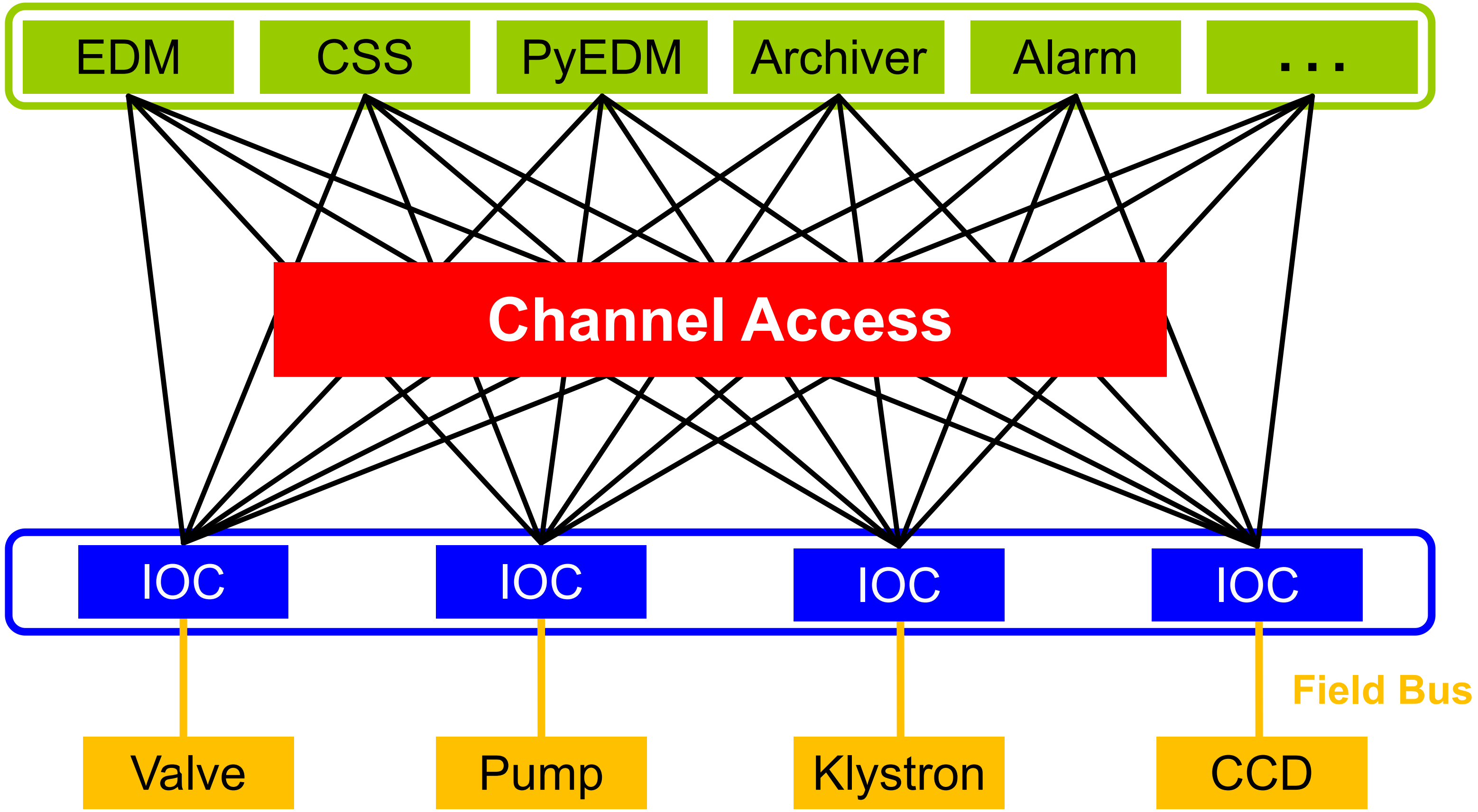
# EPICS Overview



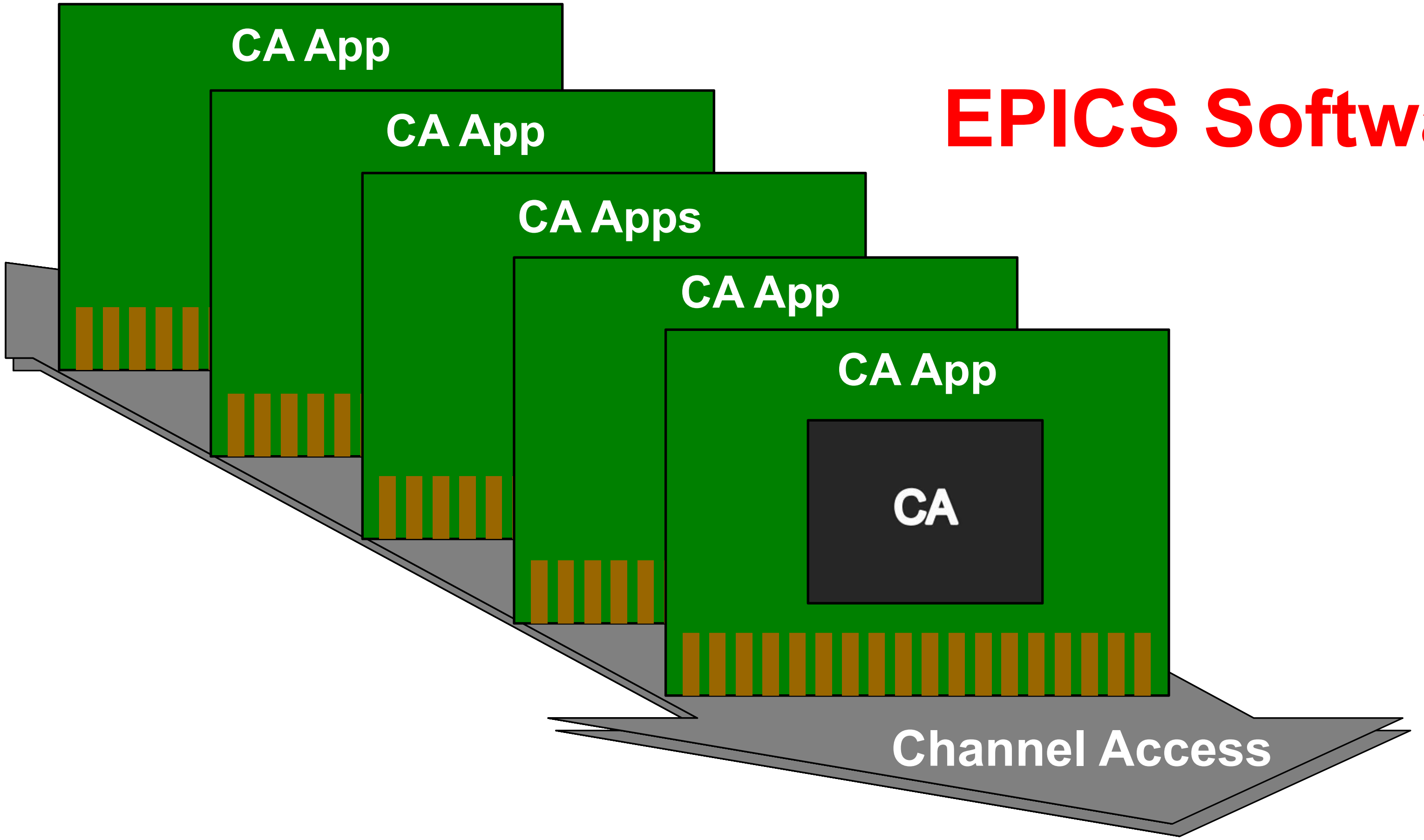
# EPICS Overview



# EPICS Overview



# Channel Access




EPICS Software Bus

PV Access：Optimized for large data (arrays, structured data)

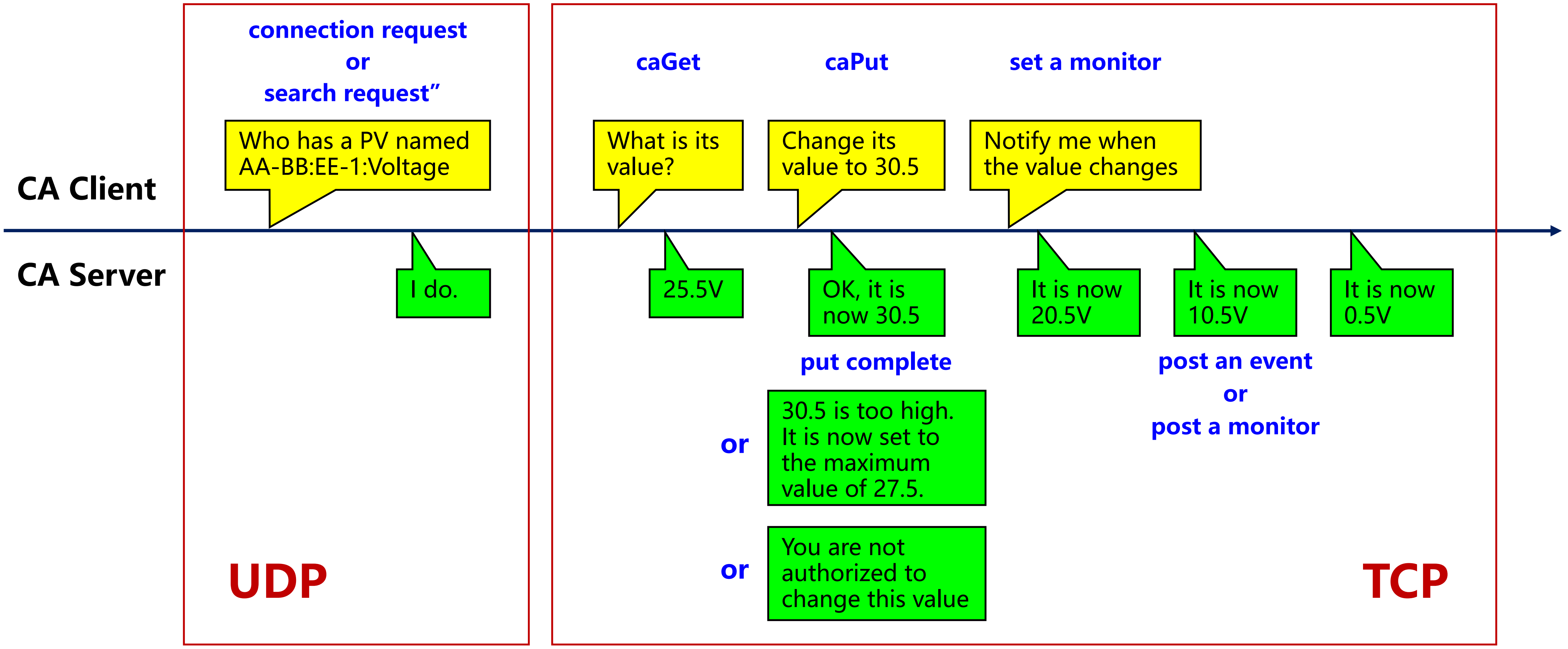
# Process Variable

- **Process Variable** (PV) is a **named item of data resides in CA/PVA server**, with optional attributes.
- **Input/Output Controller** (IOC) is the server that **hosts** PVs and **interfaces** with hardware.
- Clients can read or write PVs using PV name, which must be **unique** in control system.
- Data Types – PVs can hold different data types (integers, floats, strings, arrays, etc.).
- Attributes – PVs often include additional information like units, alarm status, and timestamp.
- Access Control – Some PVs may have read/write permissions or security restrictions.

**AC-UN-FEL1-VA:IPS-5:CURRENT-SP**

**AC-UN-FEL1-VA:IPS-5:OPEN-CMD**

# Channel Access



# CA and PVA Command line Tools

| | |
|---|---|
| caget | Reads the current value of one or more PVs. |
| caput | Writes a value to a PV. |
| camonitor | Monitors PV updates in real time. |
| cainfo | Displays detailed information about a PV (e.g., data type, host IOC). |

| | |
|---|---|
| pvget | Reads the value of one or more PVs (PVA alternative to caget). |
| pvput | Writes a value to a PV (PVA alternative to caput). |
| pvmonitor | Monitors PV changes (PVA alternative to camonitor). |
| pvinfo | Displays detailed PV metadata (PVA alternative to cainfo). |
| pvcall | Invokes a PVA RPC (Remote Procedure Call) service. |
| pvlist | Lists all PVs available on a network (PVA discovery). |

…

# CA Client Applications

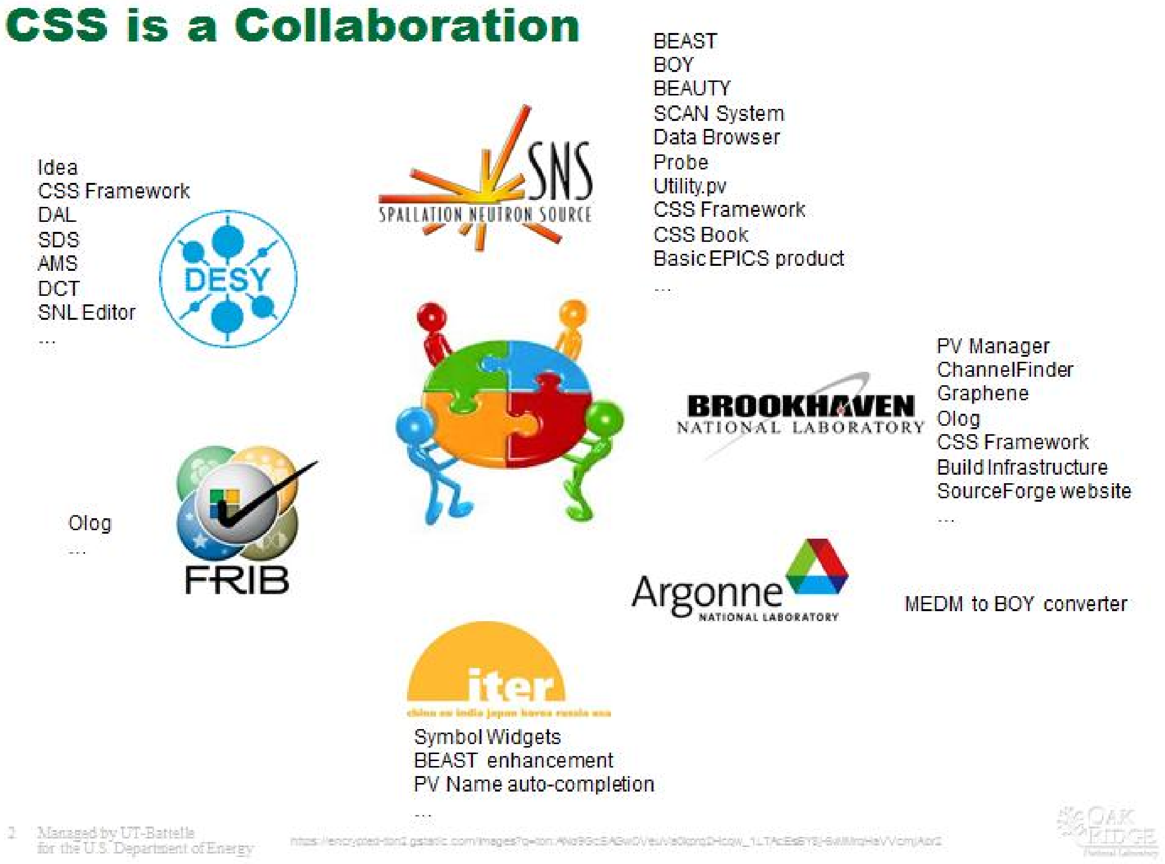
- ALH: Alarm Handler
- caQtDM: An MEDM replacement based on Qt
- CS-Studio: Control System Studio
- CS-Studio Phoebus
- EDM: Extensible Display Manager
- MEDM: Motif Editor and Display Manager
- Probe: Motif Channel Monitoring program
- PyDM: Python and PyQt-based Display Manager
- Sequencer: State Notation Compiler and Sequencer
- StripTool: Strip-chart plotting tool
- TimeChart: PyDM-based strip-chart application

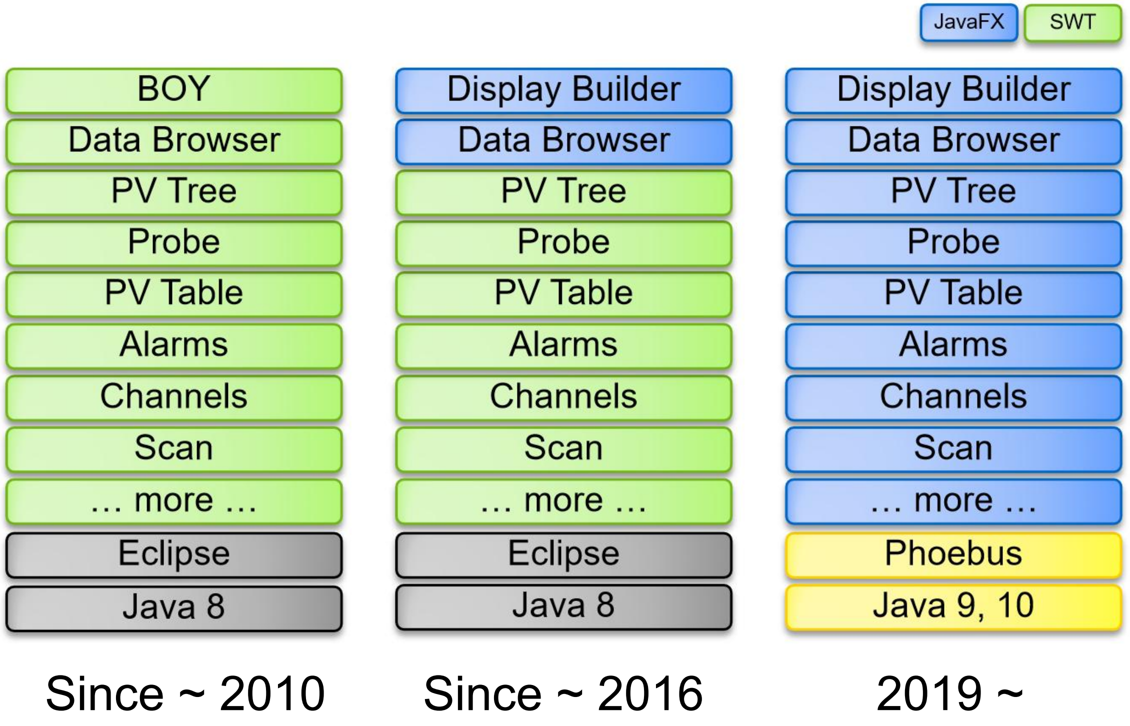
…

https://epics.anl.gov/extensions/index.php

# Control System Studio

- CSS is an Eclipse-based **collection** of tools to monitor and operate large scale control system.
- It's a product of the colaboration between diferent laboratores and universities.




Kay Kasemir, Display Builder Tutorial

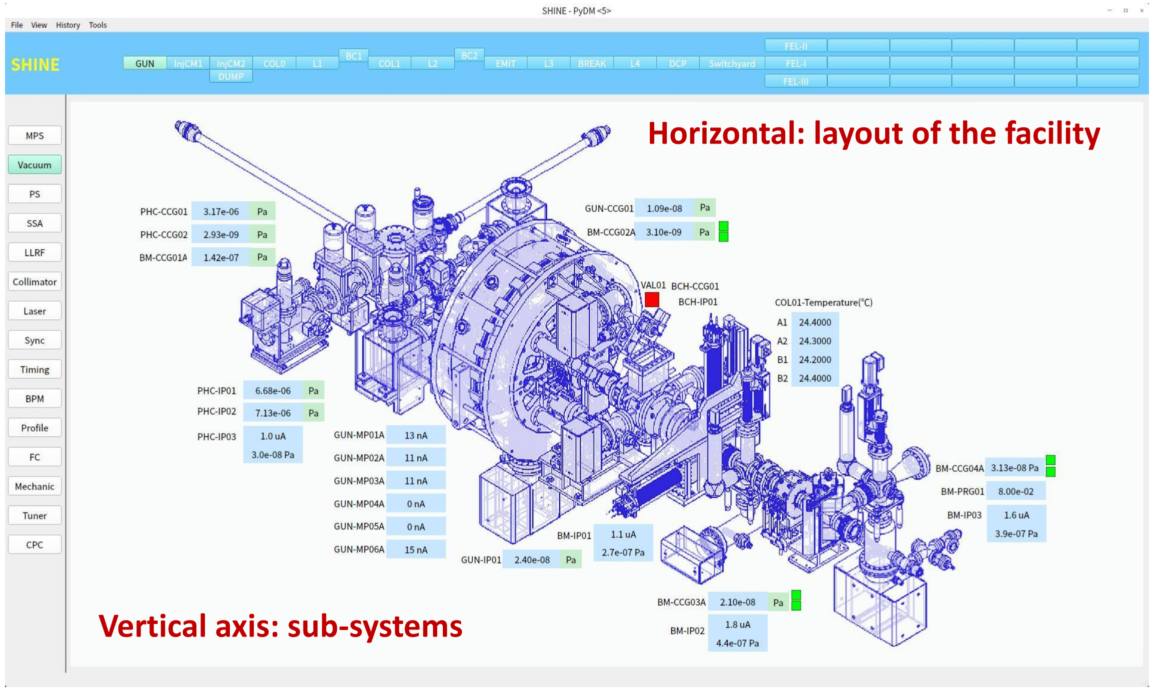
# Python Display Manager

- PyDM is a **Python-based** graphical user interface (GUI) framework for control system.
- It provides a system for the drag-and-drop creation of user interfaces using **Qt Designer**.
- It also allows for the creation of displays driven by Python.

| Widget Name | Function |
|---|---|
| PyDMPushButton | Add the toggle function |
| PyDMRelatedDisplayButton | Add the function of password authentication |
| PyDMShellCommand | Add the function of password authentication |
| PyDMPSReadbackButton | Password Button |
| ArchiverPlot | Retrieve the archiver data |
| … | |

**SHINE Customed PyDM Widgets**


https://github.com/slaclab/pydm



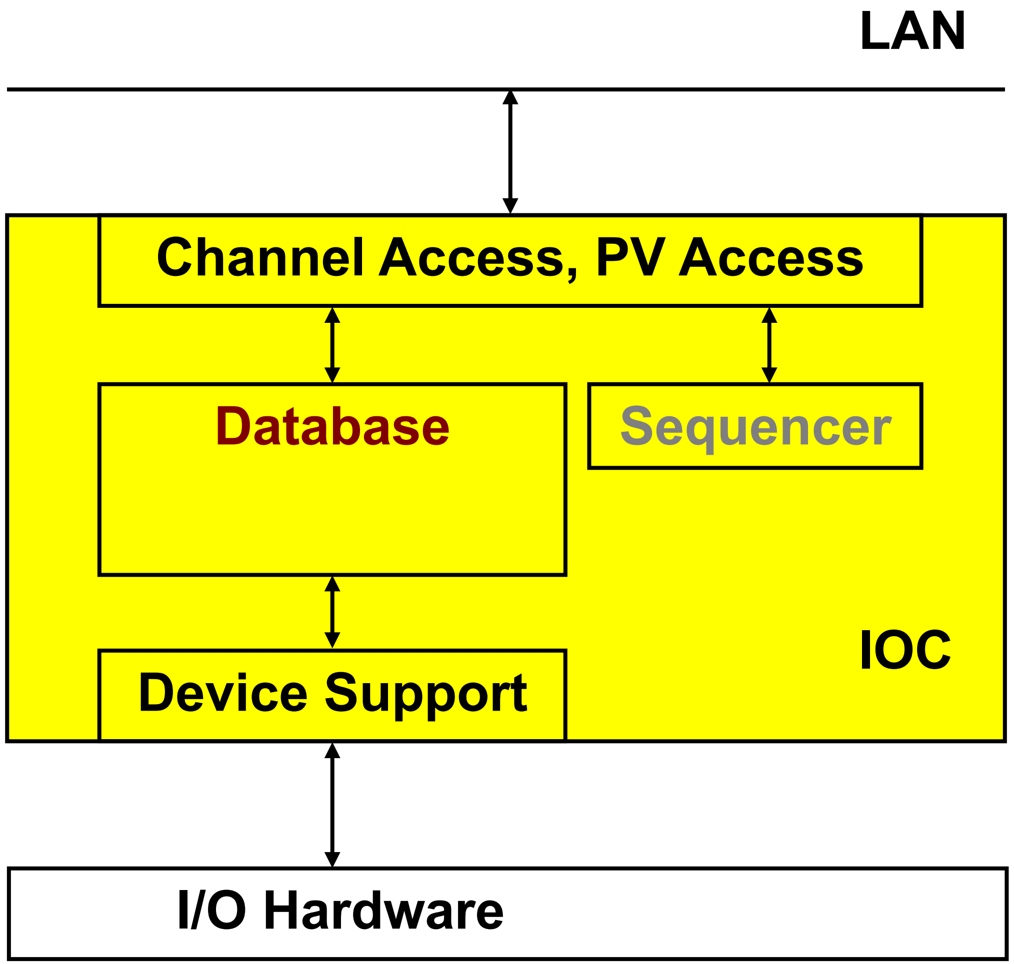
# Display Manager

| Feature | EDM | CSS | PyDM |
|---|---|---|---|
| Development Era | 1990s | 2006+ | 2016+ |
| Language | C (X11) | Java (Eclipse) | Python (Qt/PyQt) |
| Performance | Very fast (C-based) | Moderate (Java) | Fast (Qt-based) |
| EPICS Support | CA (EPICS 3) | CA & PVA (EPICS 3/7) | CA & PVA (EPICS 3/7) |
| Widget Library | Basic (limited customization) | Advanced (Eclipse plugins) | Rich (Qt widgets + Python) |
| Scripting | EDL (limited) | JavaScript, Python | Python (full flexibility) |
| Styling | Minimal | CSS-like styling | Qt Stylesheets (CSS-like) |
| Cross-Platform | Linux (X11), Windows (limited) | Windows, Linux, macOS | Windows, Linux, macOS |
| Mobile/Web Support | No | RAP | PyDMWeb, … |
| Learning Curve | Low<br>(simple but outdated) | Steep<br>(Eclipse environment) | Moderate<br>(Python/Qt knowledge helps) |

# Input/Output Controller (IOC)

- **Hardware Communication**
  - Interfaces with devices via protocols
  - Uses device support libraries (streamDevice, ...)
- **Database Processing**
  - 'iocCore' software loads and executes Records
- **Automation & Control Logic**
  - Supports sequencing (State Notation Language)
- **Process Variable (PV) Services**
  - Listens on CA or PVA for client requests.

- Database: **Records + Fields + Links**
- Sequencer: state machine, mostly on-demand



# Record

- Each record has
  - Name – **unique** on the whole network
  - Type – determines fields and their functionality
  - Fields – properties, can be read, most also written at runtime
  - Often device support to interface to hardware
  - Links to other records
- Records process **periodically** or **triggered** by events / other records
  - Get data from other records or hardware
  - Perform calculations
  - Check value ranges, raise alarms
  - Write to other records or hardware

```
record ( ai," AA-BB:EE-FF:XX") {
    field ( DESC,  "Current Readback")
    field ( DTYP , "stream")
    field ( INP,   "@xxx.proto addr")
    field ( SCAN,  ".5 second")
    field ( EGU,   "Amps")
    field ( PREC,  "2")
    field ( HIHI,  "20")
    field ( HIGH,  "18")
    field ( LOW,   "1")
    field ( LOLO,  "0")
}
```

# Record Types

| Record Type | Description | Common Use Cases |
|---|---|---|
| Analog Input (ai) | Reads analog input values (eg., voltage, temperature) | Sensors (thermocouples, ADCs) |
| Analog Output (ao) | Writes analog output values (e.g., setpoints) | Control signals (DACs, power supplies) |
| Binary Input (bi) | Reads binary/digital input (0/1) | Limit switches, status bits |
| Binary Output (bo) | Writes binary/digital output | Relays, on/off controls |
| Multi-Bit Input (mbbi) | Reads multi-bit (enum) input | Device status (e.g., 3-bit encoded states) |
| Multi-Bit Output (mbbo) | Writes multi-bit (enum) output | Mode selection (e.g., START/STOP/RESET) |
| Long Input (longin) | Reads 32-bit integer values | Counters, encoder positions |
| Long Output (longout) | Writes 32-bit integer values | Stepper motor positions |
| String Input (stringin) | Reads text strings | Device firmware versions, messages |
| String Output (stringout) | Writes text strings | Command strings (e.g., RUN=1) |
| Waveform (waveform) | Handles arrays (numeric/string) | Data buffers (e.g., spectra, images) |
| Calc (calc) | Performs math on inputs | Derived values (e.g., A+B) |
| Calcout (calcout) | Math + conditional output | Alarm thresholds, feedback loops |
| Seq (seq) | State machine | Complex sequencing (e.g., device startup) |

# Common Fields

- **Initial**
  - NAME: Record name, unique on network!
  - DESC: Description
  - SCAN: Passivem, periodically, event or I/O Intr
  - PINI: Process once on initialization?
  - FLNK: Forward link
- **Runtime**
  - TIME: Time stamp
  - SEVR, STAT: Alarm Severity, Status
  - PACT: Process active
  - UDF: Undefined? Never processed?
  - PROC: Force processing

```
record ( ai," AA-BB:EE-FF:XX") {
    field ( DESC,   "Current Readback")
    field ( DTYP ,  "stream")
    field ( INP,    "@xxx.proto addr")
    field ( SCAN,   ".5 second")
    field ( EGU,    "Amps")
    field ( PREC,   "2")
    field ( HIHI,   "20")
    field ( HIGH,   "18")
    field ( LOW,    "1")
    field ( LOLO,   "0")
}
```

# Common Fields

- Input/Output Record Fields
  - DTYP: Device type
  - INP/OUT: How to read/write, format depends on DTYP
  - RVAL: Raw value (e.g. 16 bit integer)
  - **VAL**: Engineering unit value (e.g. 64bit float)
- Output Record Fields
  - DOL: Desired Output Link.

    Output records read this link to get VAL, then write to OUT…
  - OMSL: .. if Output Mode SeLect = closed_loop
  - IVOA: Invalid Alarm Output Action
  - DRVL, DRVH: Drive limits

```
record ( ai," AA-BB:EE-FF:XX") {
    field ( DESC,  "Current Readback")
    field ( DTYP , "stream")
    field ( INP,   "@xxx.proto addr")
    field ( SCAN,  ".5 second")
    field ( EGU,   "Amps")
    field ( PREC,  "2")
    field ( HIHI,  "20")
    field ( HIGH,  "18")
    field ( LOW,   "1")
    field ( LOLO,  "0")
}
```

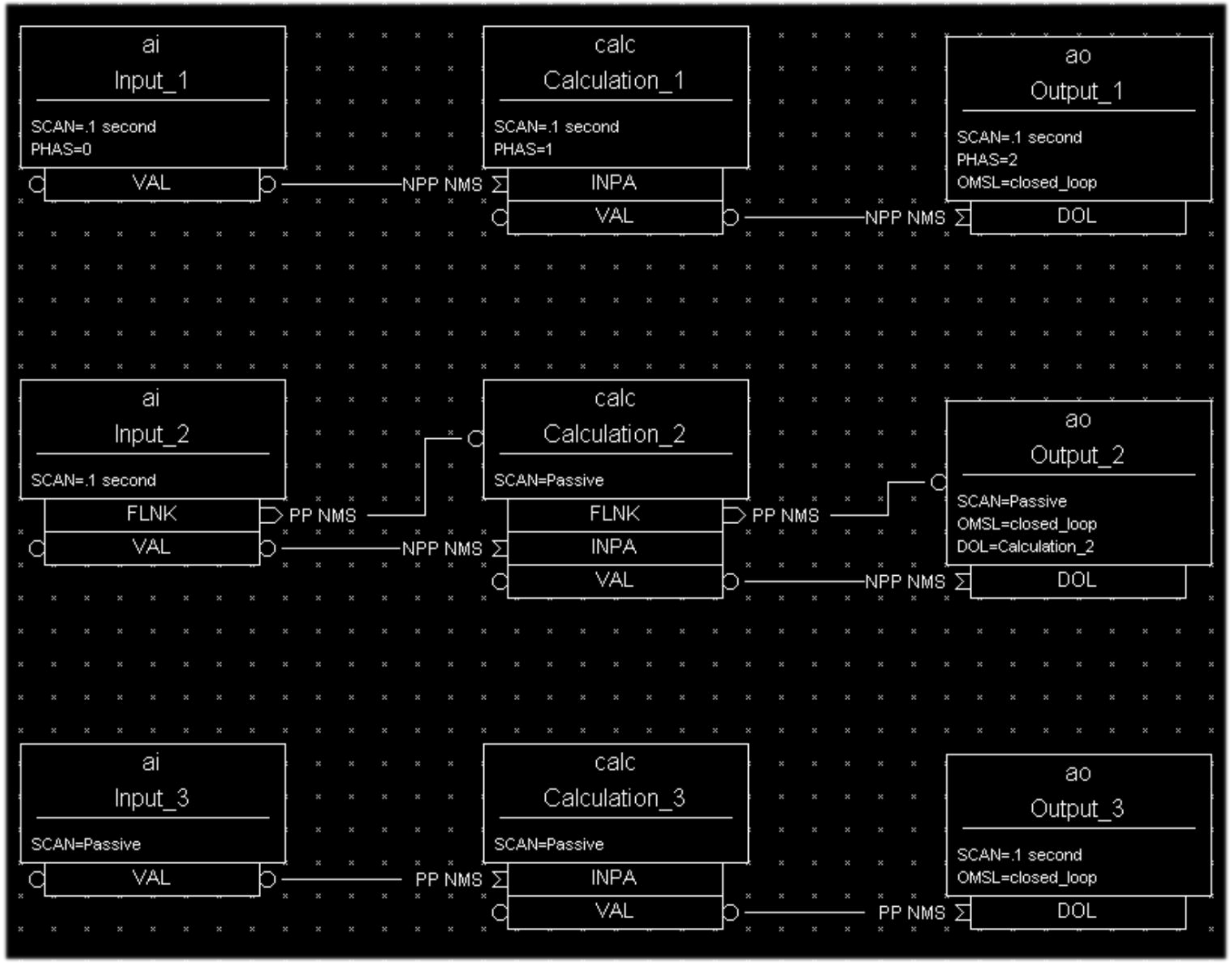
# Common Fields

- Analog Record Fields
  - EGU: Engineering units name
  - LINR: Linearization (None, Slope, breakpoint table)
  - EGUL, EGUF, ESLO, EOFF: Parameters for LINR
  - LOLO, LOW, HIGH, HIHI: Alarm Limits
  - LLSV, LSV, HSV, HHSV: Alarm severities
- Binary Record Fields
  - ZNAM, ONAM: State name for zero and one
  - ZSV, OSV: Alarm severities

```
record ( ai," AA-BB:EE-FF:XX") {
    field ( DESC,   "Current Readback")
    field ( DTYP ,  "stream")
    field ( INP,    "@xxx.proto addr")
    field ( SCAN,   ".5 second")
    field ( EGU,    "Amps")
    field ( PREC,   "2")
    field ( HIHI,   "20")
    field ( HIGH,   "18")
    field ( LOW,    "1")
    field ( LOLO,   "0")
}
```

# Database Links

- PP: Process a passive target record
  - INP, DOL: Before reading
  - OUT: After writing
- NPP: non-process-passive (**default**)
- MS: Maximize severity
- NMS: non-MS (default)
- MSS: Maximize Severity and Status
- MSI: .. when severity = INVALID



# Access Security

- A networked control system must have the ability to enforce **security rules**
  - Who can do what from where, and when?
- In EPICS, security is enforced by the **CA server** (the IOC or **gateway**)
- A record is placed in the **Access Security Group** named in its ASG field
  - DEFAULT is used if no group name is given
- Rules are specified for each group to determine whether a CA client can read or write to records in that group, based on
  - Client user ID
  - Client host-name or IP address
  - Access Security Level of the field addressed
  - Values read from the database

```
UAG(users) {user1, user2}
HAG(hosts) {host1, host2}
ASG(DEFAULT) {
    RULE(1, READ)
    RULE(1, WRITE) {
            UAG(users)
            HAG(hosts)
    }
}
```

Access Security Configuration File

# Hardware and Software Support

- A hardware support module provides software for use within an IOC to control a real-world commercial device.
- A soft support module may contain a new record type, software-only device support, or some other software that runs in the IOC but which is not readily identified with a particular piece of hardware.

Total Records Found: 676, showing 50 per page

| Manufacturer | Bus | Module | Description | Contact | Link |
|---|---|---|---|---|---|
| | ARM IOC (Linux) | GPIO | /dev/class/gpio on BeagleBone Black, Raspberry Pi etc. | Florian Feldbauer | GH:devgpio |
| | Firewire (Asyn) | IIDC DCAM cameras | AreaDetector plug-in to control and read out IIDC DCAM compliant firewire cameras | Ulrik Pedersen | DLS:DCAM |
| | IEEE1394 (Asyn) | IIDC DCAM cameras | AreaDetector plug-in to control and read out IIDC DCAM compliant firewire cameras | Ulrik Pedersen | DLS:DCAM |
| | IOC-Arduino (Streams) | | Raspberry Pi to Arduino via USB | Pete Jemian | GH:cmd_response |
| | Modbus (Asyn) | PLCs | Modbus TCP, serial RTU and serial ASCII | Mark Rivers | CARS:Modbus |
| | USB (Asyn) | TMC | Test and Measurement Class devices | Eric Norum | APS:asyn |

25 entries per page — Search:

| Class | Name | Description | Contact | Link |
|---|---|---|---|---|
| Driver (Asyn) | iocshDev | iocsh command device support | Eric Norum (wenorum@lbl.gov) | APS:modules |
| embedded linux | Build rootfs | Scripts to build a minimal Linux rootfs | Michael Abbott (michael.abbott@diamond.ac.uk) | Diamond:Rootfs |
| facility | areaDetector | Support for cameras etc. Uses ASYN | Mark Rivers (rivers@cars.uchicago.edu) | GH:areaDetector |
| facility | ASYN | General-purpose device interface facility | Mark Rivers (rivers@cars.uchicago.edu) | APS:asyn |
| facility | asynPythonDriver | An asynPortDriver in Python | Wang Xiaoqiang (xiaoqiang.wang@psi.ch) | github:asynPythonDriver |
| facility | asynUSBTMC | ASYN support for USB Test & Measurement Class devices | Eric Norum (wenorum@lbl.gov) | SF:asynUSBTMC |
| facility | Autosave | Bumpless IOC Reboot support | Tim Mooney (mooney@aps.anl.gov) | APS:synApps/autosave |
| facility | caPutLog | Logging of CA puts to iocLogServer | Benjamin Franksen (benjamin.franksen@bessy.de) | BESSY:caPutLog |

https://epics-controls.org/resources-and-support/modules/soft-support/

# streamDevice

- streamDevice is a generic device support module that simplifies communication with devices using **text-based protocols** (e.g., SCPI, custom ASCII commands).
- Based on AsynDriver, it translates record operations into **device-specific command sequences**.
- **Serial** (RS-232, RS422, RS485), **Network** (TCP, UDP), **GPIB** (IEEE-488)

**Protocol File**

```
Terminator = CR LF;
InTerminator = LF;
ReplyTimeout = 10000;
ReadTimeout = 10000;
getTempA
{
   out "KRDG? A";
   in "%f";
}
```

**DB File**

```
record(ai, "Temp:A")
{
   field( DESC, "Temperature Readback")
   field(DTYP, "stream")
   field(INP, "demo.proto getTempA L0")
   field(SCAN,"5 second")
   field( EGU, "...")
   ...
}
```

**IOC st.cmd**

```
drvAsynSerialPortConfigure("L0","/dev/tty.X",0,0,0)
asynSetOption("L0", -1, "baud", "9600")
asynSetOption("L0", -1, "bits", "8")
asynSetOption("L0", -1, "parity", "none")
asynSetOption("L0", -1, "stop", "2")
asynSetOption("L0", -1, "clocal", "Y")
asynSetOption("L0", -1, "crtscts", "Y")
...
```

https://epics.web.psi.ch/software/streamdevice/

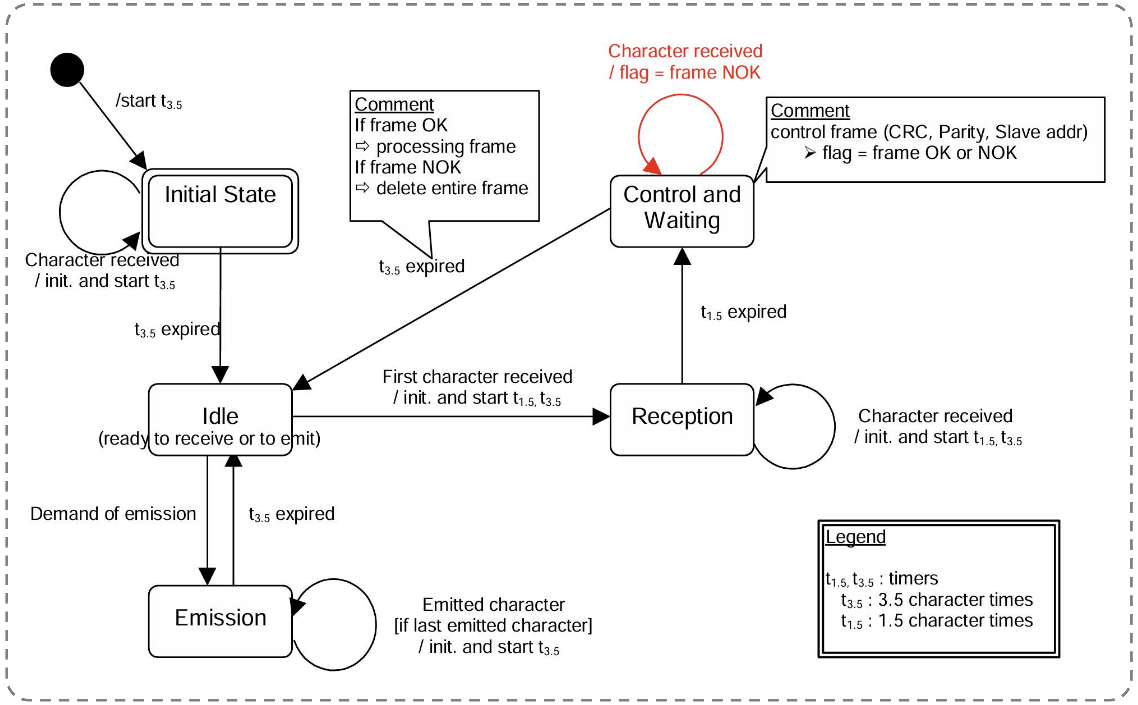
# Modbus

- The Modbus module is a software component that allows the IOC to communicate with devices using the **Modbus protocol**, which is widely used in industrial automation systems.
- Supports 3 communication-link layers: **TCP**, **RTU** and **Serial ASCII**.
- Provides access to the following 4 types of data.

| Primary tables | Object type | Access |
|---|---|---|
| Discrete Inputs | Single bit | Read-Only |
| Coils | Single bit | Read-Write |
| Input Registers | 16-bit word | Read-Only |
| Holding Registers | 16-bit word | Read-Write |

| Slave Address | Function Code | Data | CRC |
|---|---|---|---|
| 1 byte | 1 byte | 0 up to 252 byte(s) | 2 bytes<br>CRC Low, CRC Hi |

RTU Message Frame

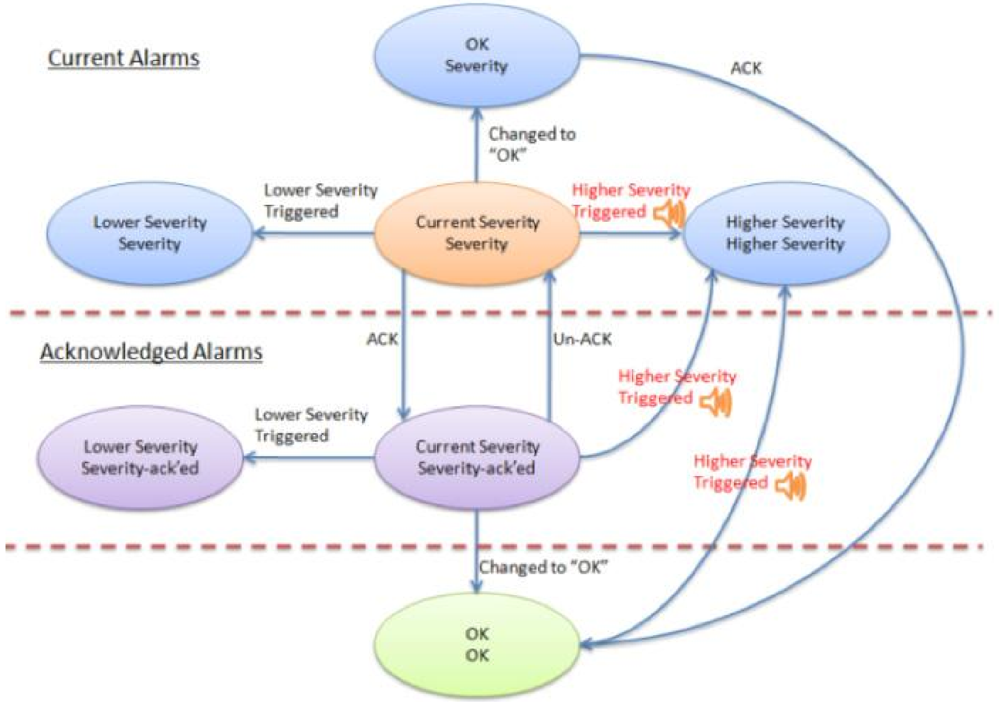



RTU Transmission Mode State

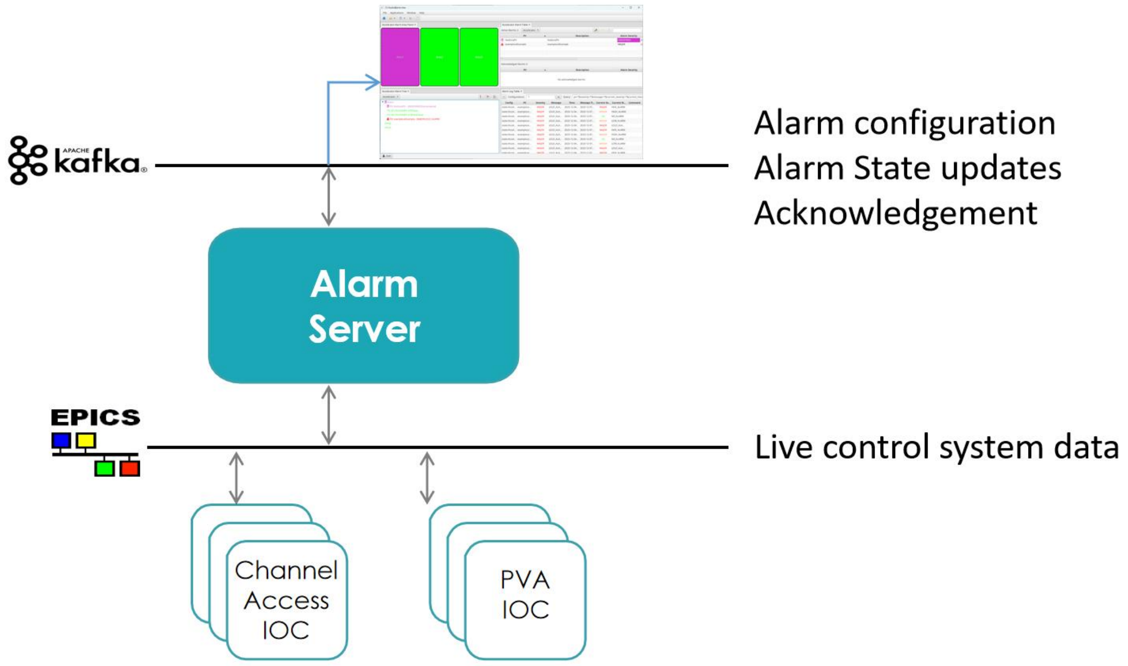
https://epics.anl.gov/extensions/index.php

# Alarm

- The Alarm is a subsystem within EPICS that **monitors** and **manages** alarms for process variables.
- It helps operators detect and respond to abnormal conditions in the large-scale control system.
- Alarm conditions can be configured using **alarm thresholds** (e.g., HIHI, HIGH, LOW, LOLO).
- Alarm can trigger actions (e.g., notifications, logging, or automated responses).



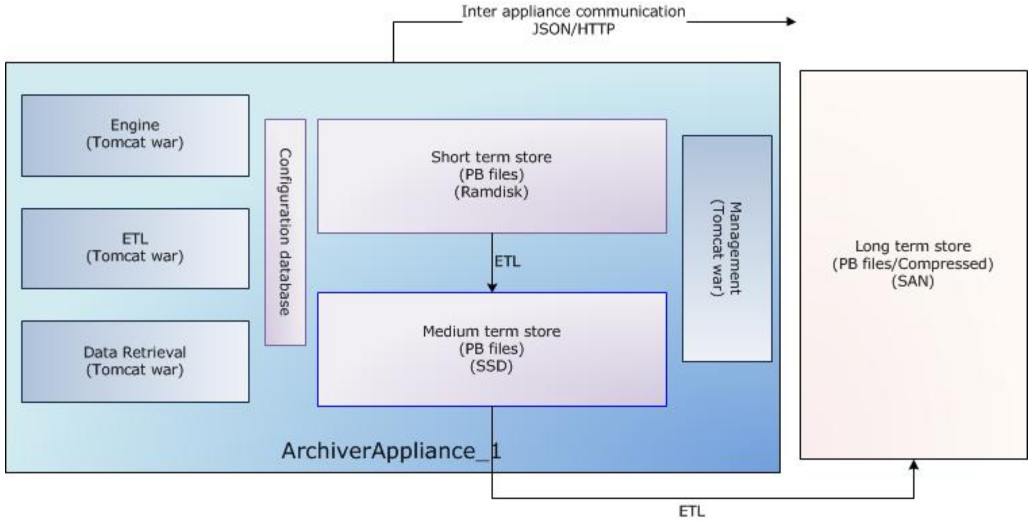

Kay Kasemir, The best ever alarm system toolkit, ICALEPCS 2009

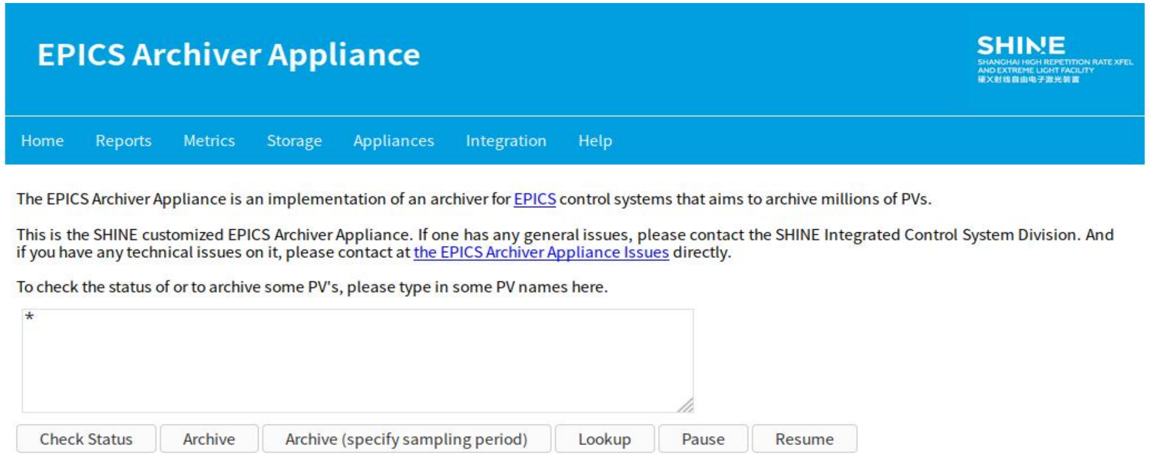


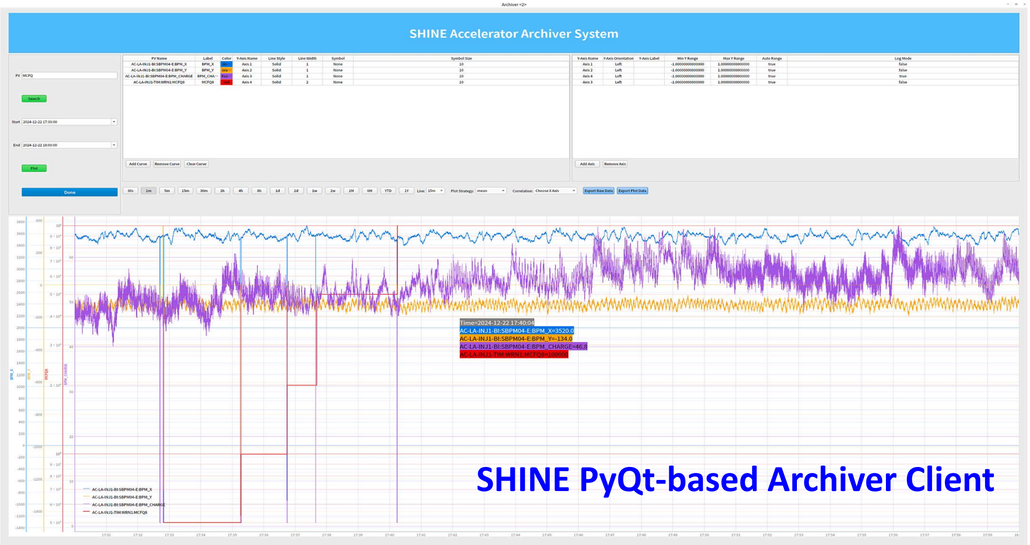

https://cs-studio.sourceforge.net/docbook/ch14.html

# Archiver

- The **Archiver Appliance** is an implementation of an archiver that aims to archive millions of PVs.
- Ability to **cluster** appliances and to scale by adding appliances to the cluster.
- Multiple stages and an inbuilt process to move data between the stages.
- Focus on data retrieval performance.





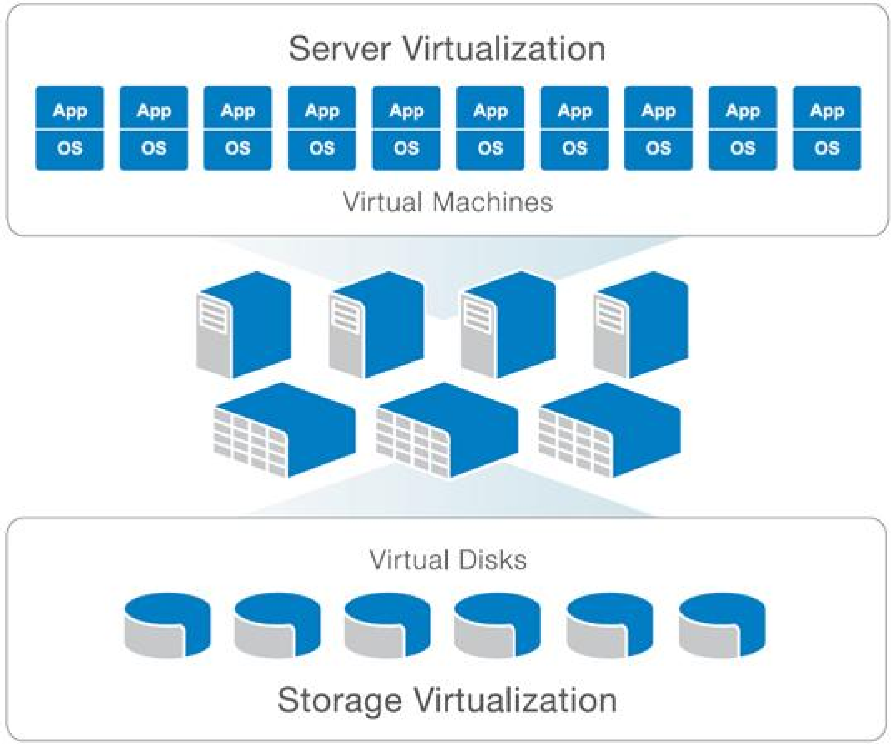

https://epicsarchiver.readthedocs.io/en/latest/

# Infrastructure & Services

# Virtualization Servers

- Virtualization servers use software (hypervisors) to create and manage **multiple virtual machines** (VMs) on the physical servers.
- Each VM runs its **own** OS and applications, sharing the host's CPU, memory, and storage.
- Maximizes HW utilization and improves system **stability**.
- KVM, VMware, Xen, Proxmox VE, Sangfor, …

| Feature | Virtualization Server | Docker |
|---|---|---|
| Technology | Hypervisor-based | Container Engine |
| Isolation Level | OS-level isolation | Process-level isolation (shared OS kernel) |
| Performance | Higher overhead (runs full OS) | Near-native (lightweight) |
| Startup Time | Slow (seconds to minutes) | Fast (milliseconds) |
| Use Case | Running multiple OSes | Microservices, cloud-native apps |
| Networking | Complex (virtual switches, VLANs) | Simplified (Docker network bridges) |
| Orchestration | vSphere, Proxmox clusters | Kubernetes, Docker Swarm |

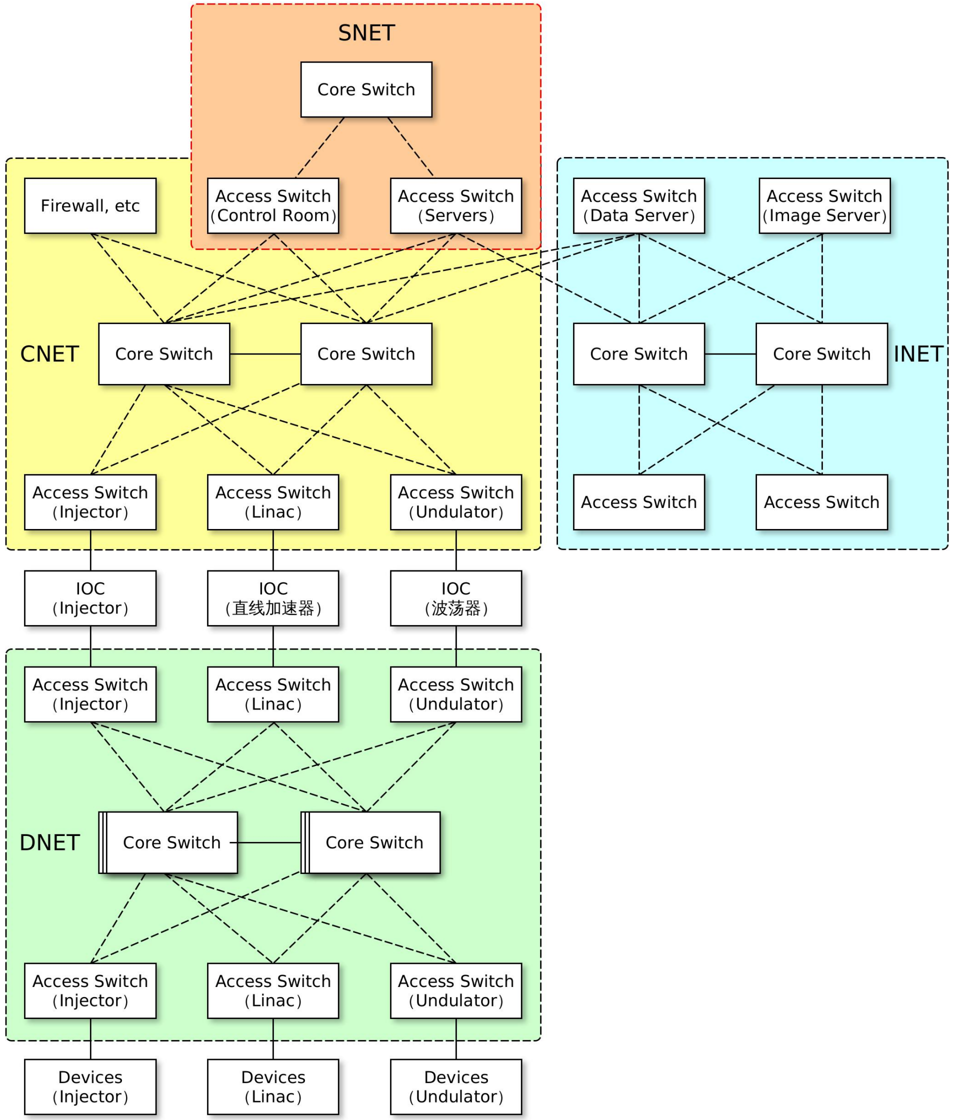


# SHINE Control Network

- **CNET** : Control NETwork for OPIs, IOCs and servers.
- **DNET** : Device dedicated NETwork for IOCs and devices.
- **INET** : Image dedicated NETwork for CCDs and servers.
- **SNET** : Storage and console dedicated NETwork.

- All networks are physically isolated.
- Redundant network of double core fiber switches.
- Subnet managed by **VLAN**.
- 100Gigabit backbone, Gigabit to access equipment.

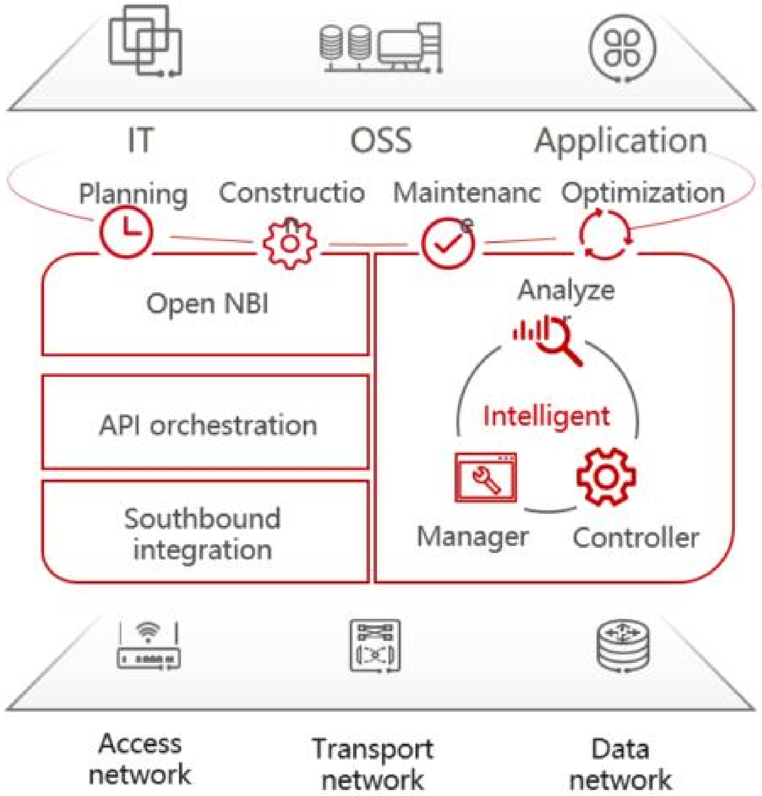
- Core Switch : HUAWEI S12700E-8
- Access Switch : HUAWEI S5736-S4874XC ~ 300

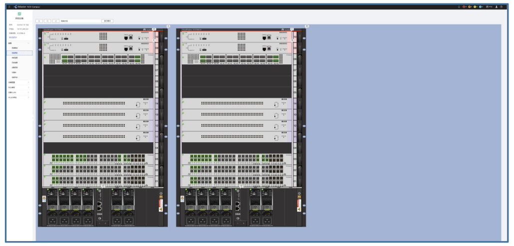


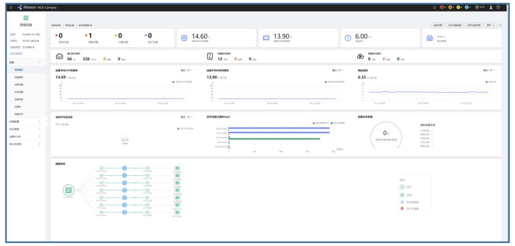
# Network Management

- **Network Management System** (NMS) is a set of applications designed to **monitor**, **analyze**, and **manage** the network infrastructure (switches, routers, IoT devices,

| Name | Vendor | License |
|---|---|---|
| iMaster NCE | Huawei | Commercial |
| SolarWinds NPM | SolarWinds | Commercial |
| Cisco DNA Center | Cisco | Commercial |
| PRTG Network Monitor | Paessler | Freemium (100 free) |
| Zabbix | - | Open-source |
| Nagios Core | - | Open-source |
| Prometheus + Grafana | - | Open-source |
| | … | |

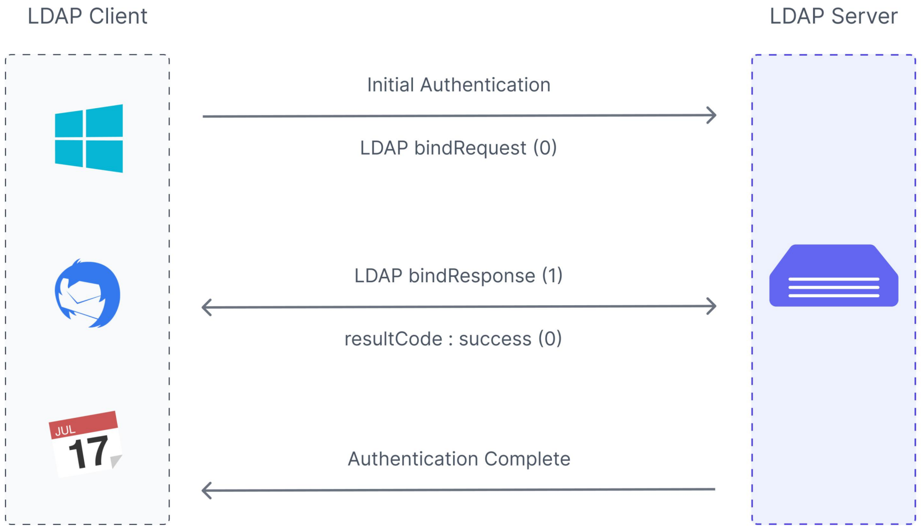




# LDAP

- LDAP (**Lightweight Directory Access Protocol**) is an **open-source** application protocol that allows applications to access and **authenticate** specific user information across directory services.



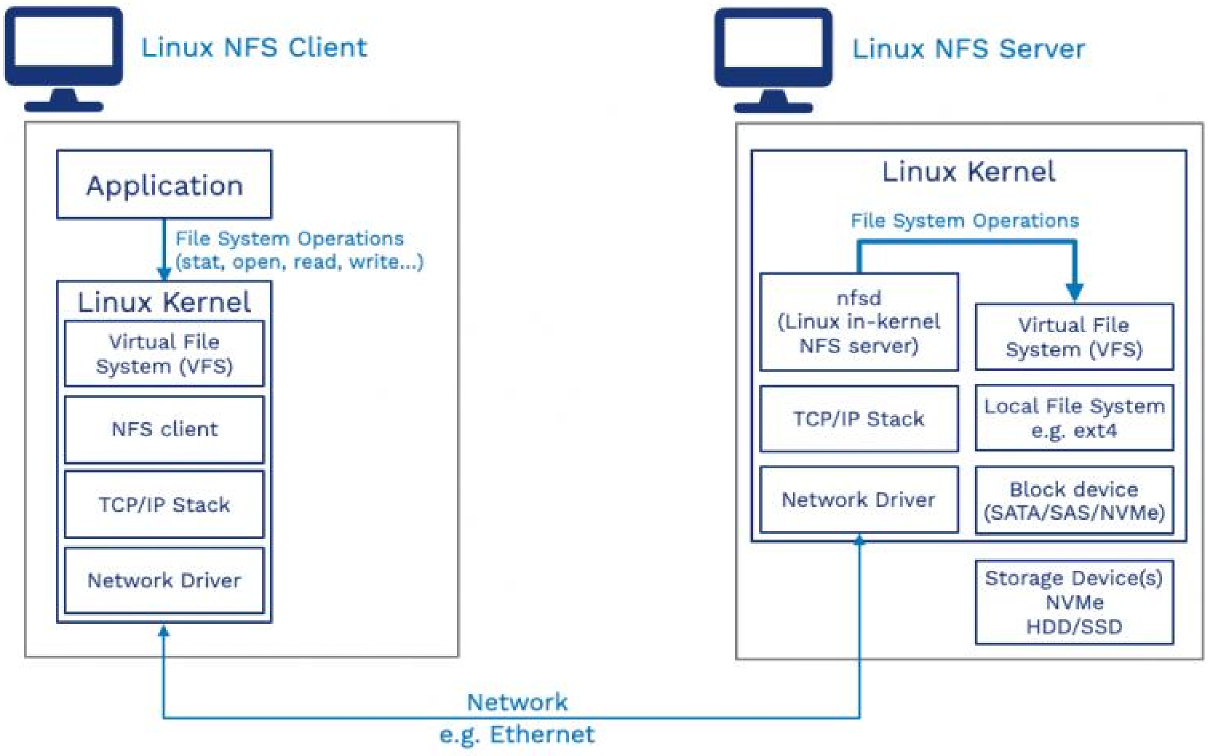

https://adaptive.live/blog/authentication-protocols-types-and-uses

| Feature | LDAP<br>Lightweight Directory Access Protocol | NIS+<br>Network Information Service Plus |
|---|---|---|
| Architecture | Hierarchical<br>(tree-based, flexible schema) | Tabular<br>(with improved structure over NIS) |
| Protocol | TCP/IP<br>(LDAP on 389, LDAPS on 636) | RPC<br>(with encryption support, port 111) |
| Security | Strong<br>(TLS/SSL, Kerberos, SASL, ACLs) | Better than NIS (DES encryption, secure RPC) but weaker than LDAP |
| Scalability | Highly scalable<br>(supports millions of entries) | Limited<br>(better than NIS but not enterprise-grade) |
| Authentication | Modern<br>(PAM, SSO, Active Directory integration) | Unix-centric<br>(improved over NIS but still niche) |
| Flexibility | Highly flexible<br>(custom schemas, multi-vendor support) | Fixed schema<br>(but more structured than NIS) |
| Performance | Optimized for reads/writes<br>in distributed systems | Faster than LDAP in small networks<br>but less efficient at scale |

# NFS

- NFS (**Network File System**) is a distributed file system protocol that allows users to access and share files over a network as if they were stored locally.
- It is widely used in Unix/Linux environments for seamless file sharing between servers and clients.

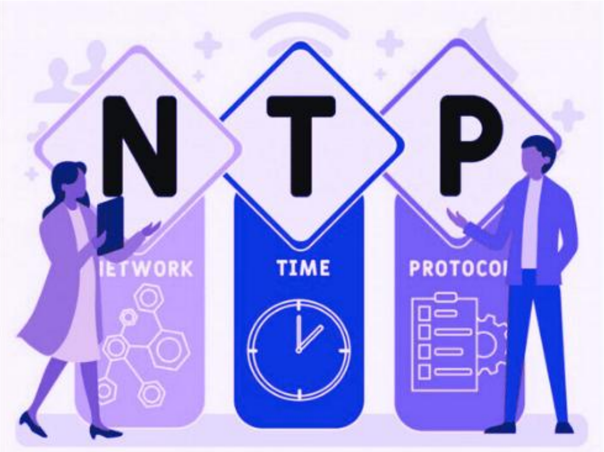
| Feature | NFS | SMB/CIFS (Windows) | GlusterFS (Distributed) |
|---|---|---|---|
| Protocol | RPC (UDP/TCP) | SMB (TCP) | Custom (TCP) |
| Security | IP-based (v3) Kerberos (v4) | NTLM Kerberos | SSL POSIX ACLs |
| Performance | Good for Unix workloads | Optimized for Windows | Scalable for large data |
| Use Case | Linux/Unix environments | Windows networks | Cloud/ container storage |

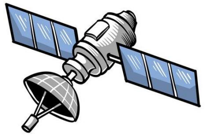


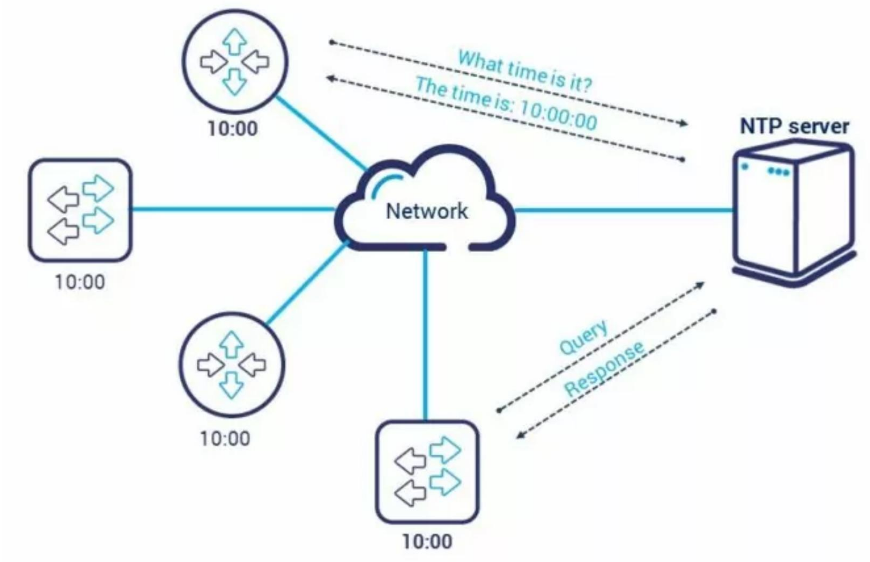

https://www.quobyte.com/storage-explained/what-is-network-filesystem/

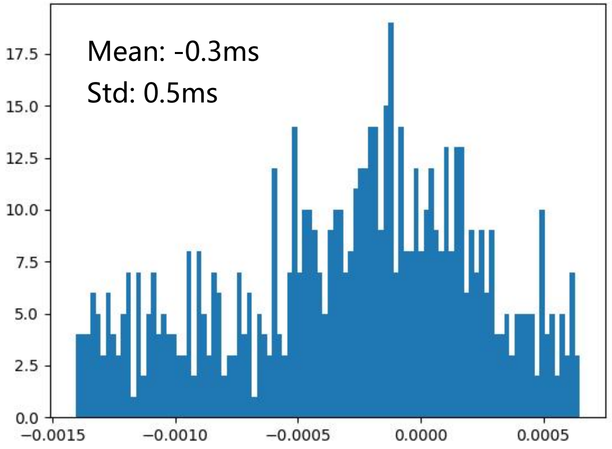
# NTP

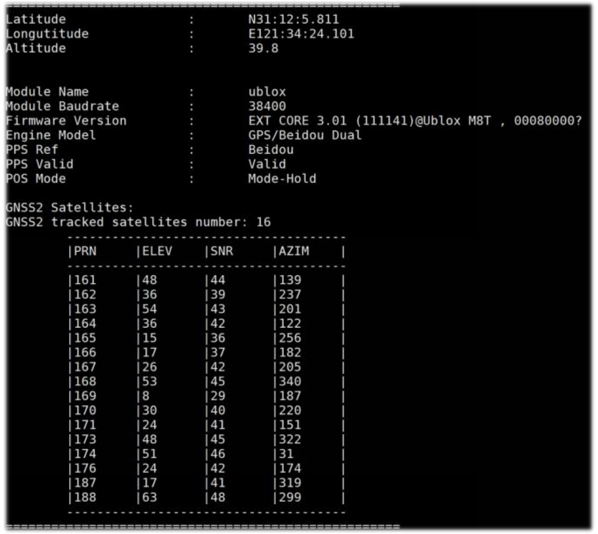
- NTP (**Network Time Protocol**) is a networking protocol used to synchronize the **clocks** of computers and devices over a network.
- Chrony is a modern time synchronization **tool** designed as an alternative to the traditional NTP daemon (ntpd).





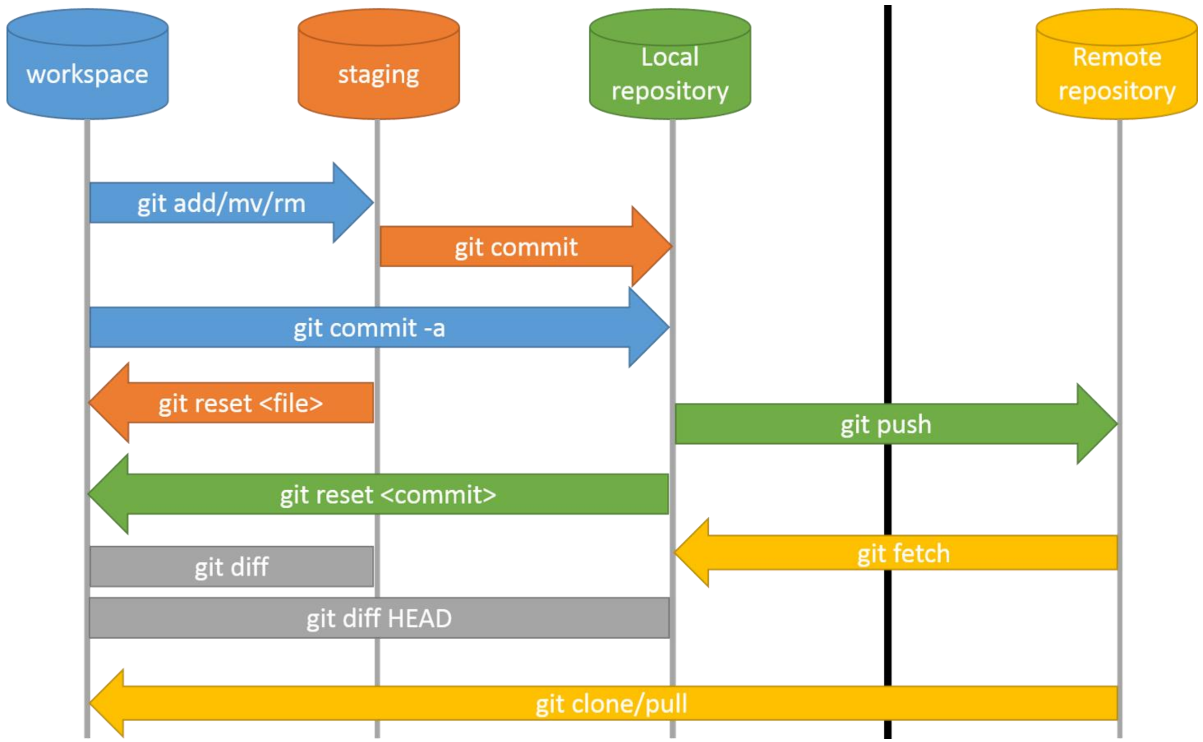



```
Latitude              :       N31:12:5.811
Longutitude           :       E121:34:24.101
Altitude              :       39.8

Module Name           :       ublox
Module Baudrate       :       38400
Firmware Version      :       EXT CORE 3.01 (111141)@Ublox M8T , 00080000?
Engine Model          :       GPS/Beidou Dual
PPS Ref               :       Beidou
PPS Valid             :       Valid
POS Mode              :       Mode-Hold

GNSS2 Satellites:
GNSS2 tracked satellites number: 16
        ------------------------------------
        |PRN     |ELEV    |SNR     |AZIM    |
        ------------------------------------
        |161     |48      |44      |139     |
        |162     |36      |39      |237     |
        |163     |54      |43      |201     |
        |164     |36      |42      |122     |
        |165     |15      |36      |256     |
        |166     |17      |37      |182     |
        |167     |26      |42      |205     |
        |168     |53      |45      |340     |
        |169     |8       |29      |187     |
        |170     |30      |40      |220     |
        |171     |24      |41      |151     |
        |173     |48      |45      |322     |
        |174     |51      |46      |31      |
        |176     |24      |42      |174     |
        |187     |17      |41      |319     |
        |188     |63      |48      |299     |
        ------------------------------------
```

https://chrony-project.org/ https://engineeringfactsz.com/what-is-ntp/

# VCS

- **Git** is a distributed **version control system** (**VCS**). It helps you keep track of code changes, collaborate with other developers, and manage different versions of your codebase.
- **GitLab** is a platform that enhances Git with collaboration, automation, and DevOps features.



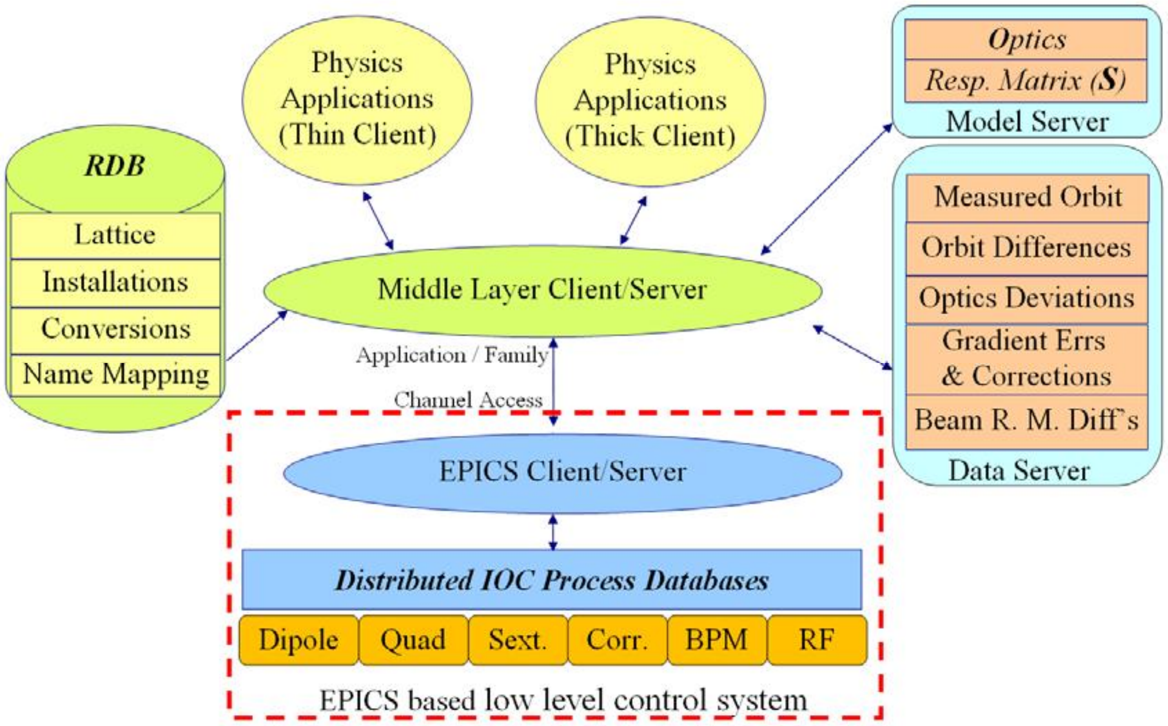

| Feature | GitLab | GitHub |
|---|---|---|
| Free Private Repos | ☑ | ☑ (now) |
| CI/CD Built-in | ☑ | ☒ |
| Self-Hosting | ☑ | ☒ (except GitHub Enterprise) |
| Community Size | Smaller | Larger |

git GitLab GitHub

# High-Level Applications

- HLAs are software tools used for beam optimization, control, and physics modeling.
- **AT (Accelerator Toolbox)**
  - Matlab-based library for accelerator physics modeling and beam dynamics simulations.
  - Developed by ESRF, now widely used.
- **Open XAL**
  - Java/Python framework for online accelerator physics applications developed by ORNL.
- **LOCO** (Linear Optics from Closed Orbits)
  - Correct accelerator optics using BPM data.
  - Used at LCLS, NSLS-II, DLS, SSRF, ...

...

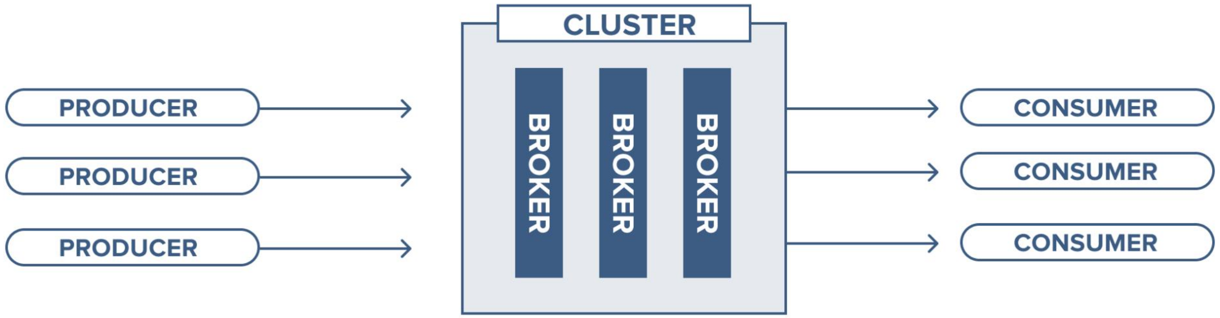


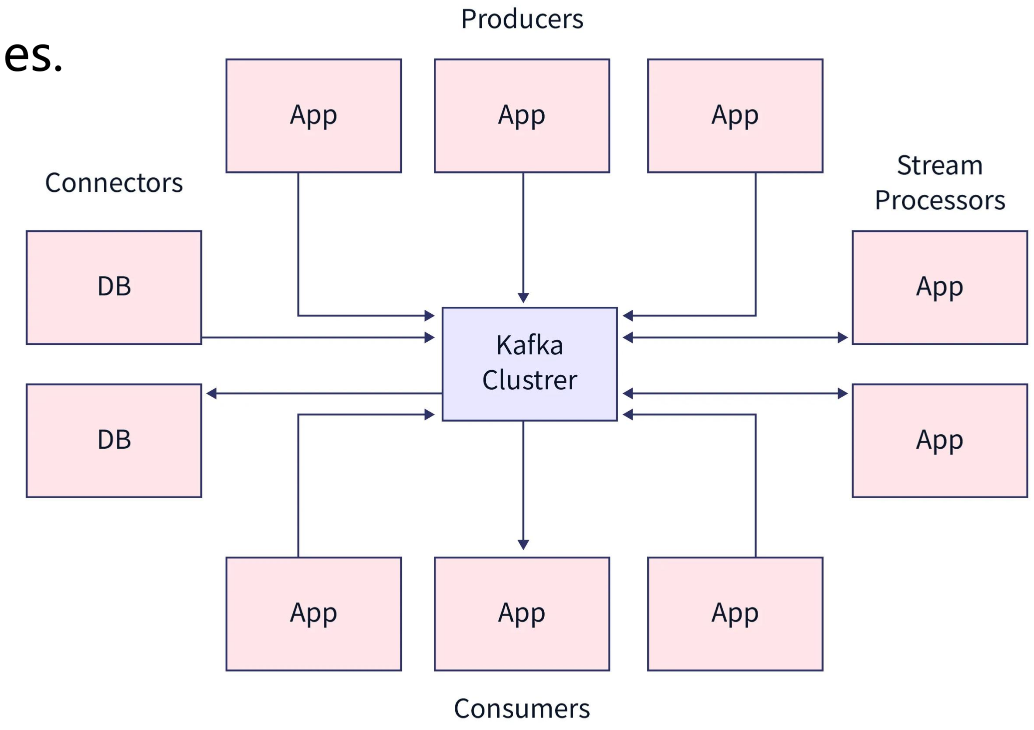

NSLS-II High Level Application Architecture (G.B. Shen)

# Kafka

- Apache Kafka is an **open-source**, distributed data store optimized for ingesting and processing **streaming data** in real-time.
- It was originally developed by LinkedIn and later donated to the Apache Software Foundation.
- High Throughput – Handles millions of messages per second.
- Fault Tolerance – Replicates data across multiple nodes.
- Low Latency – Delivers messages in milliseconds.

kafka



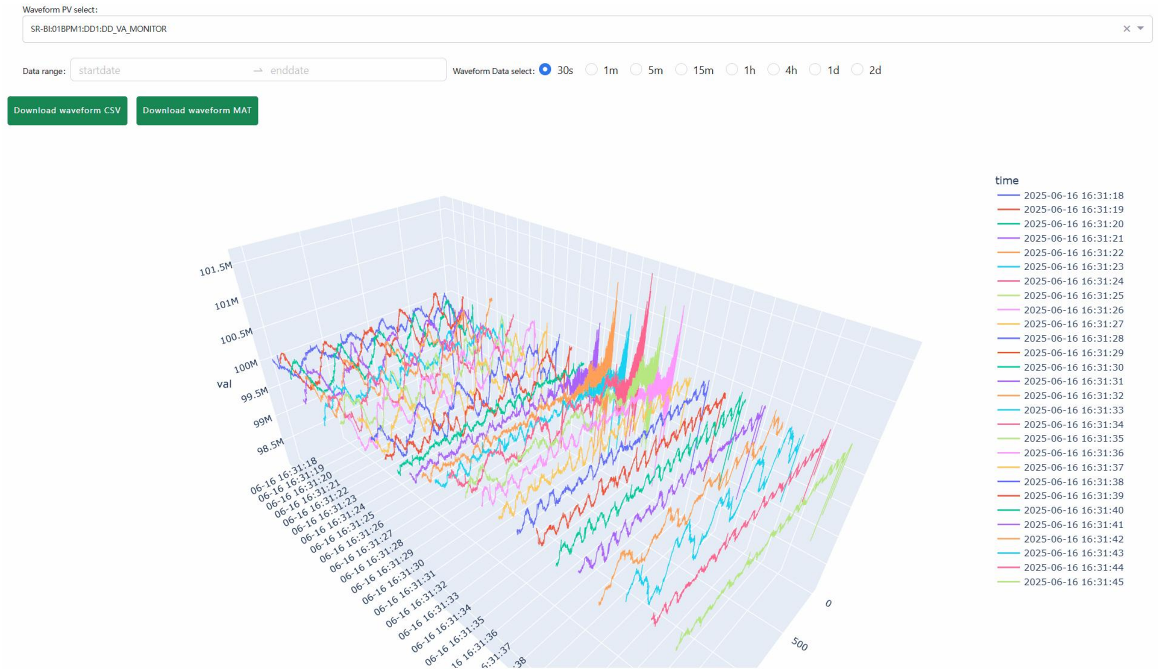
# ZeroMQ

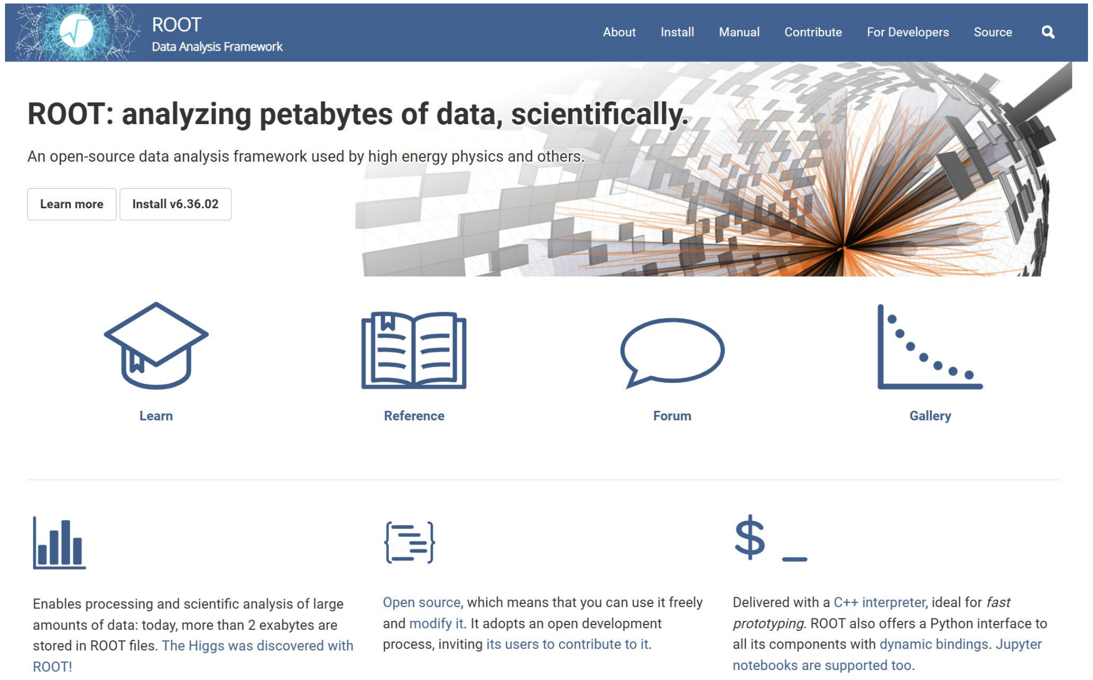
- **ZeroMQ (ZMQ)** is a high-performance, **asynchronous messaging library** designed for distributed and concurrent applications.
- Brokerless Architecture – Direct peer-to-peer communication.
- Multiple Transport Protocols – Supports TCP, IPC, WebSocket, multicast.
- Flexible Messaging Patterns – Pub/Sub, Request/Reply, Pipeline, etc.
- Lightweight – Tiny footprint (~50KB core library).

| Feature | Apache Kafka | ZeroMQ | RabbitMQ | ActiveMQ |
| --- | --- | --- | --- | --- |
| Architecture | Distributed log (brokered) | Brokerless (P2P sockets) | Centralized broker | Centralized broker |
| Messaging Model | Pub/Sub + Stream Processing | Pub/Sub, Req/Rep, Pipeline, etc. | Pub/Sub, Queues (AMQP) | Pub/Sub, Queues (JMS/AMQP) |
| Latency | ms to sec (batch optimized) | µs to ms (lowest) | ms | ms |
| Use Cases | Event streaming, log aggregation | Microservices, HFT, IoT | Task queues, workflows | Enterprise messaging |

# Data Storage and Analysis

- Accelerators generate **massive volumes of data** from control systems, beam diagnostics and detectors.
- Efficient data storage and analysis require specialized architectures and tools.
- Data Storage: Archiver Appliance, HDB++, HDF5, ROOT, MongoDB, ...
- Data Analysis: JupyterHub, MadAnalysis5, Psana, TensorFlow, PyTorch, ...

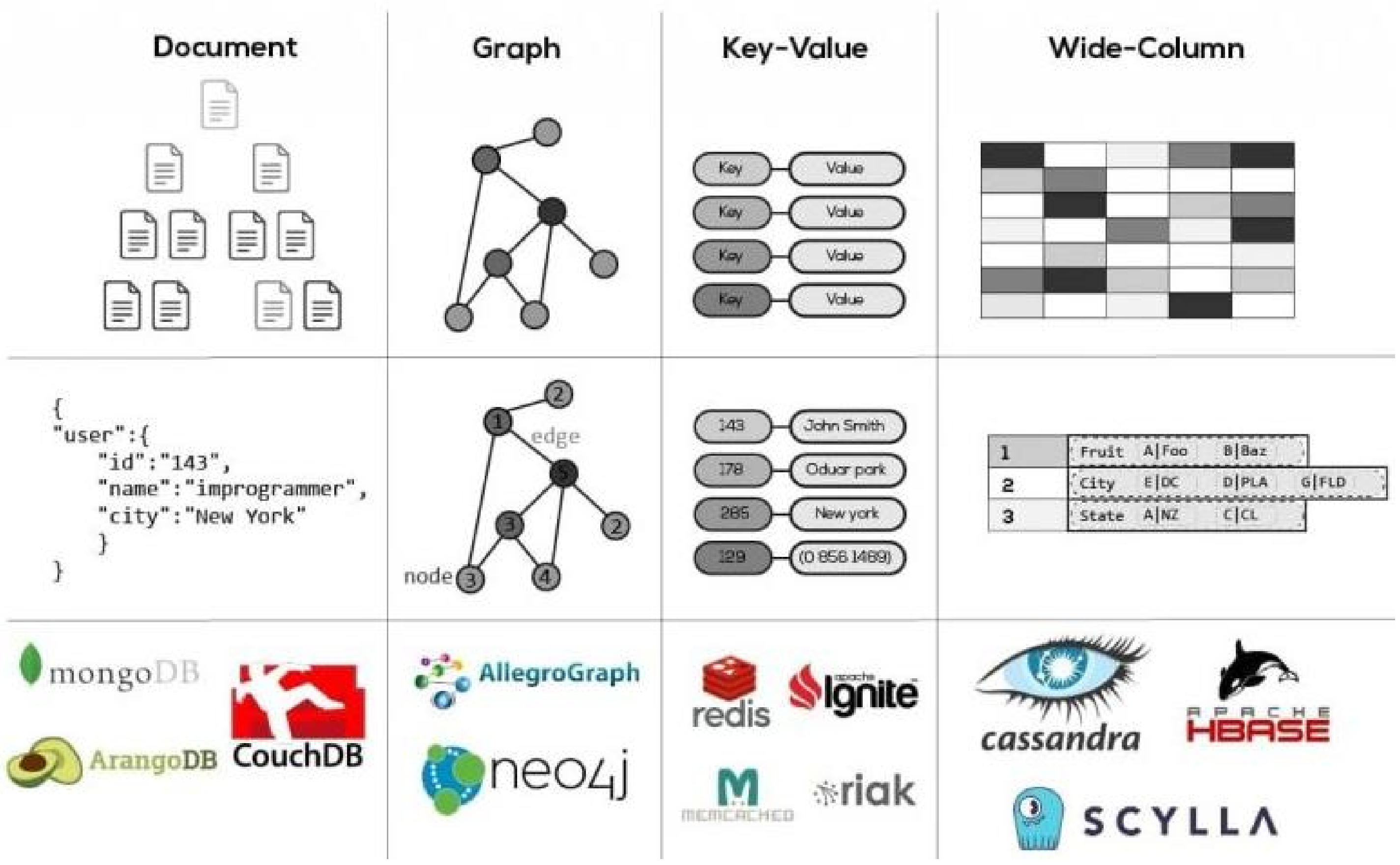



https://root.cern.ch/

# NoSQL Database

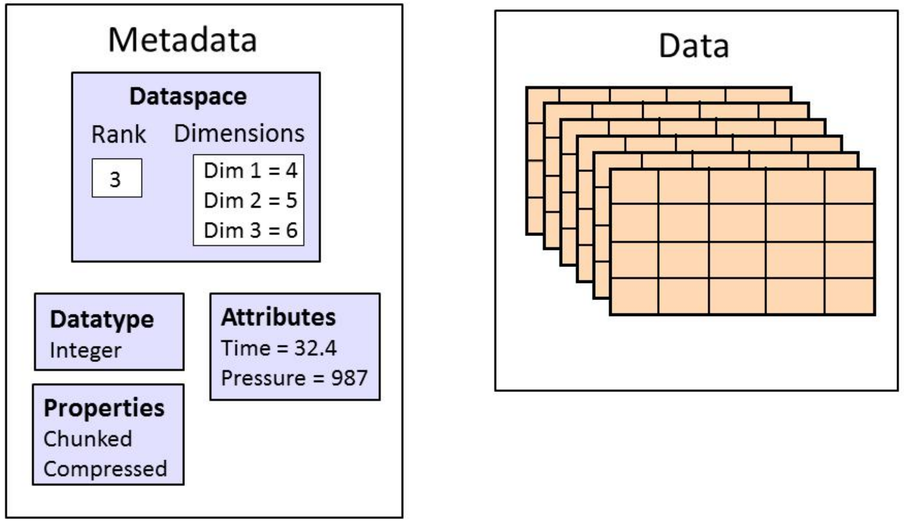
- NoSQL (**Not Only SQL**) is a type of database management system designed to handle and store large volumes of **unstructured** and **semi-structured** data.
- Unlike traditional relational databases that use tables with pre-defined schemas, NoSQL databases use **flexible data models** that can adapt to changes in data structures and scale horizontally to handle growing amounts of data.

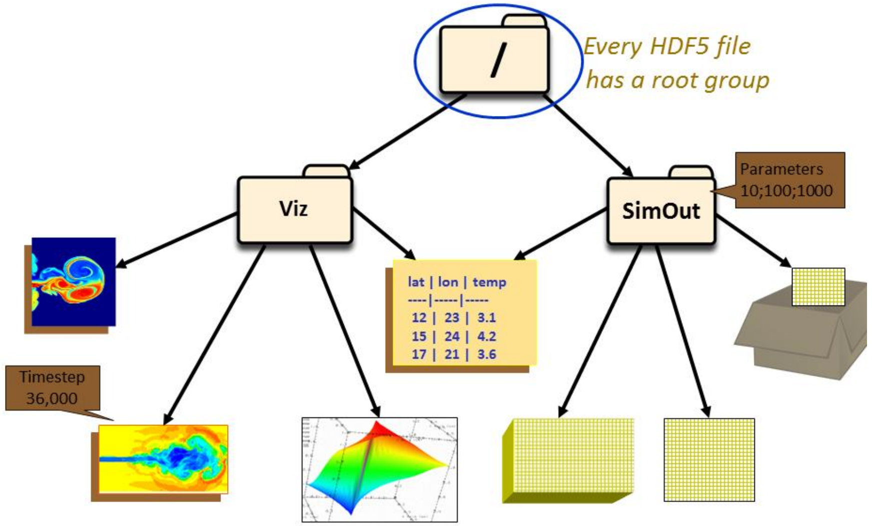



https://miska.co.in/how-do-nosql-databases-work/

# HDF5

- HDF5 (**Hierarchical Data Format version 5**) is an **open-source** file format and data model designed for storing and managing large, complex datasets efficiently.
- It is widely used in scientific computing, big data analytics, and high-performance computing due to its flexibility, scalability, and support for heterogeneous data.
- HDF5 file can be thought of as a **container** that holds a variety of heterogeneous data **objects** (or **datasets**).
- A dataset consists of **metadata** that describes the data, in addition to the data itself.




https://support.hdfgroup.org/documentation/hdf5/latest/_intro_h_d_f5.html

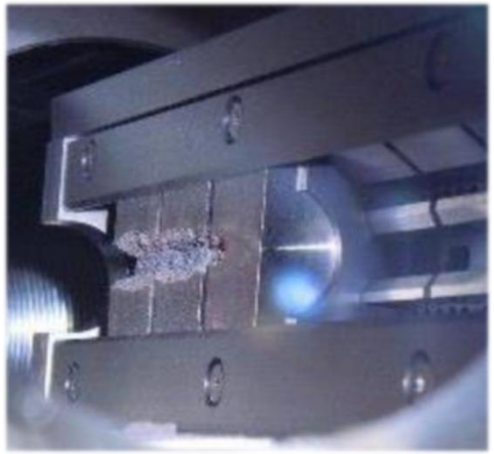
# Machine Protection

**Some materials and ideas are copied from Jörg Wenninger and Markus Zerlauth (CERN)**

# Definition

Accelerators, as all other technical systems, must respect some general principles with respect to safety.

- Protect the people (legal requirements).
- Protect the environment (legal requirements).
- Protect the equipment (asset management).
  - Without beam : superconducting magnets, RF systems, etc.
  - **With beam: damage caused by beams.**
- Those 3 aspects may be coupled in some circumstances!



**Machine Protection is focused on equipment protection from damage caused by beams.**

# Risks and Protection

- Protection is required since there is some risk.

Risk = **Probability** of an accident

x **Consequences** (money, downtime, radiation doses)

- Probability of an uncontrolled beam loss:
  - What are the failures that lead to beam loss into equipment?
  - What is the probability for the failure modes?
- Consequences:
  - Damage to equipment.
  - Downtime of the accelerator for repair.
  - Activation of material, dose to personne.

| | | | | | |
|---|---|---|---|---|---|
| 5 Catastrophic | 1 | 2 | 3 | 4 | 5 |
| 4 Major | 0 | 1 | 2 | 3 | 4 |
| 3 Severe | 0 | 0 | 1 | 2 | 3 |
| 2 Minor | 0 | 0 | 0 | 1 | 2 |
| 1 Slight | 0 | 0 | 0 | 0 | 1 |
| | A 1/10000 Years | B 1/1000 Years | C 1/100 Years | D 1/10 Years | E 1/1 Year |

**The higher the risk, the more protection becomes important !**

# Objectives ($P^3$)

- **Protect the machine**
  - Highest priority is to avoid damage of the accelerator.

- **Protect the beam**
  - Complex protection systems reduce the availability of the accelerator, the number of 'false' interlocks stopping operation must be minimized.
  - Trade-off between protection and operation.

- **Provide the evidence**
  - Clear (post-mortem) diagnostics must be provided when:
    - the protection systems stop operation,
    - something goes wrong (failure, damage, but also 'near miss')

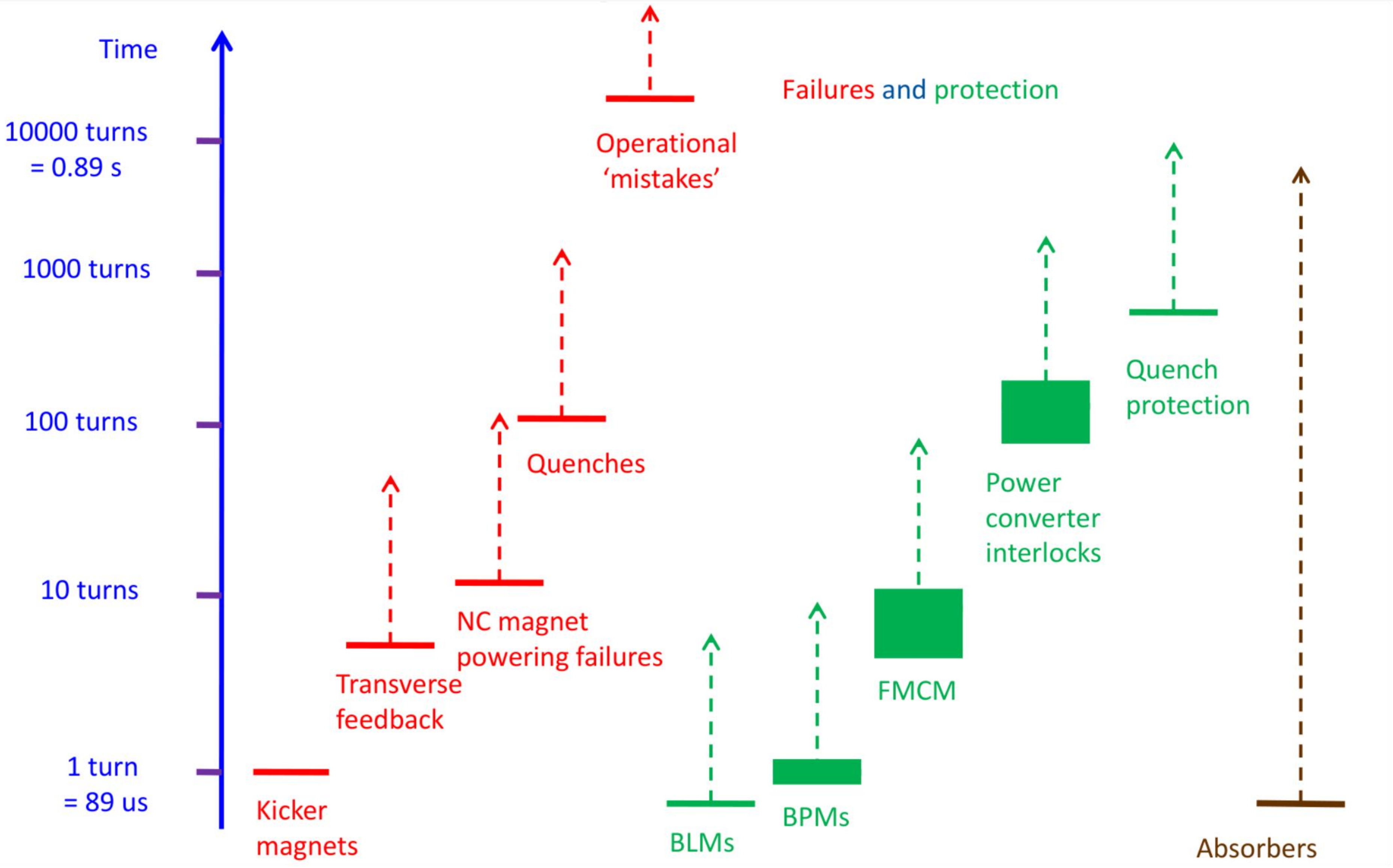
# Failure Classification

- Failure type:
  - **Hardware failure** (AC distribution failure, vacuum leak, RF trip, ...).
  - **Controls failure** (wrong data, trigger problem, feedback failure, ...).
  - **Operational failure** (chromaticity / tune / orbit errors, ...).
  - **Beam instability** (high beam / bunch current).
- Failure parameters:
  - **Damage** potential.
  - **Probability** for the failure.
  - **Time constant** for beam loss.
- Machine state (when failure occurs):
  - Linac, beam transfer, injection and extraction (single pass).
  - Stored beam.

# Design Strategy

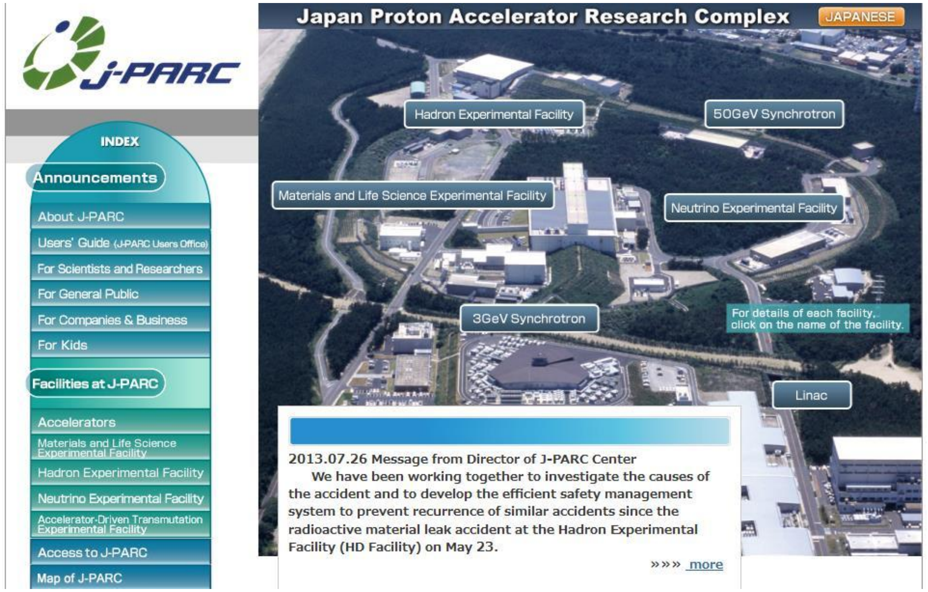
- Avoid a failure by design – if you can.
- **Detect a failure at the hardware** (equipment) level and stop operation. – 1st protection layer
- **Detect the consequences of the failure** on beam parameters (orbit, tune, losses etc) and stop operation. – 2nd protection layer
- Stop beam operation
  - Inhibit injection,
  - Send beam to a dump,
  - Stop the beam by collimators / absorbers.
- Elements of protection:
  - Equipment and beam monitoring,
  - Collimators and absorbers,
  - Beam dumps,
  - Interlock system linking different systems.

# Timescales @ LHC



# Failure Analysis

- **Figure out what can go wrong…**
- Requires good understanding of accelerator physics: how does a given element affect the beam?
- Requires good understanding of the hardware: time scales, failure modes?
- Requires a complete overview of all machine equipment that affect the beam.
- The analysis must be done systematically for every system, from bottom up – including the software/controls.




- Due to a power converter failure, a slow extraction was transformed into a fast extraction.

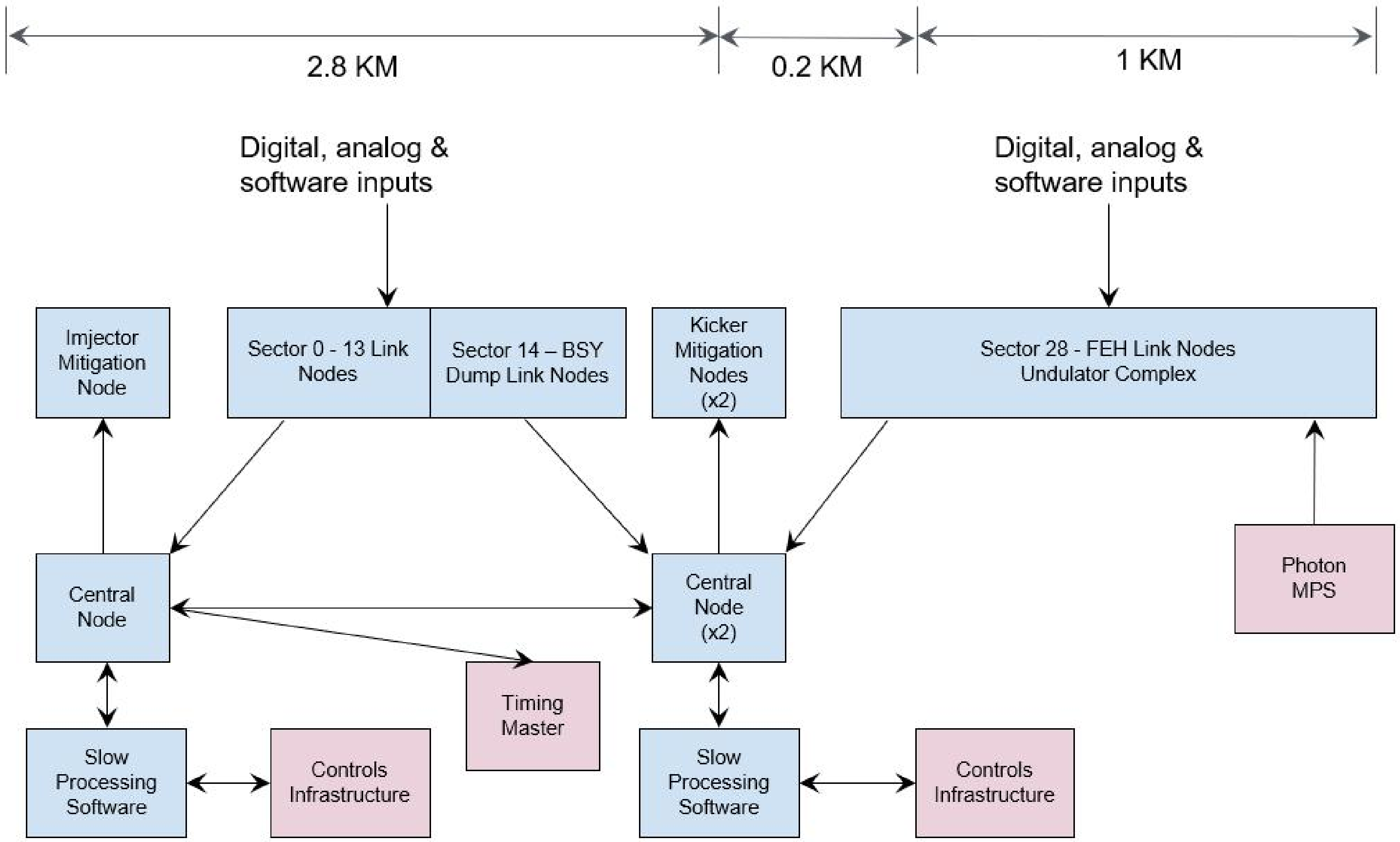
  **Extraction in milliseconds instead of seconds.**
- As a consequence of the high peak power a target was damaged and radio-isotopes were released into experimental halls.

  **Machine protection coupled to personnel protection !**

**One insufficiently covered failure case had major consequences !**

# Architecture

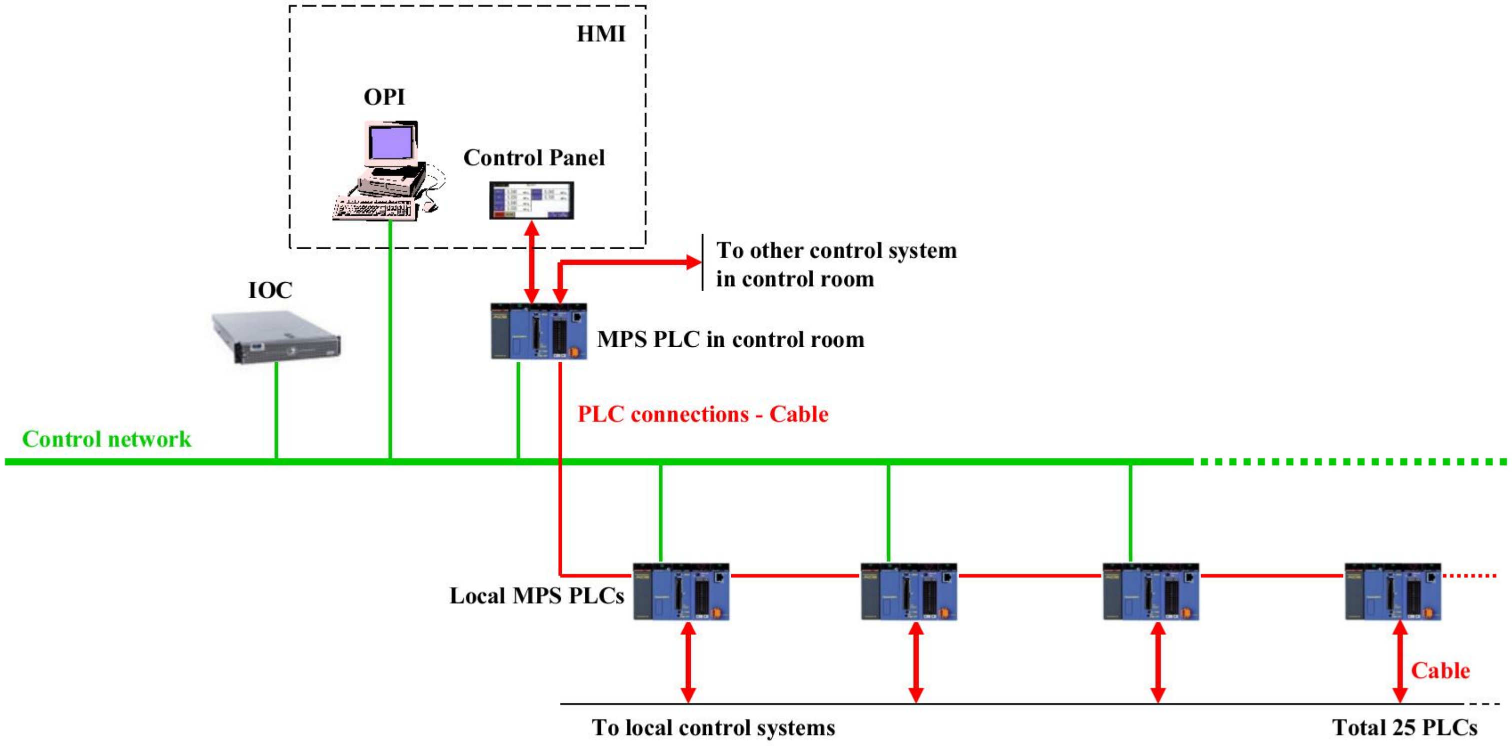
- MPS is comprised of a collection of distributed nodes that collect and process data and send them off the central node.
- The central node collects the data and compares them against a pre-programmed logic table to determine the overall state of the accelerator.
- The output is then distributed to the destination nodes for certain action.




J. A. Mock, Commissioning of the LCLS-II Machine Protection System for MHz CW Beams

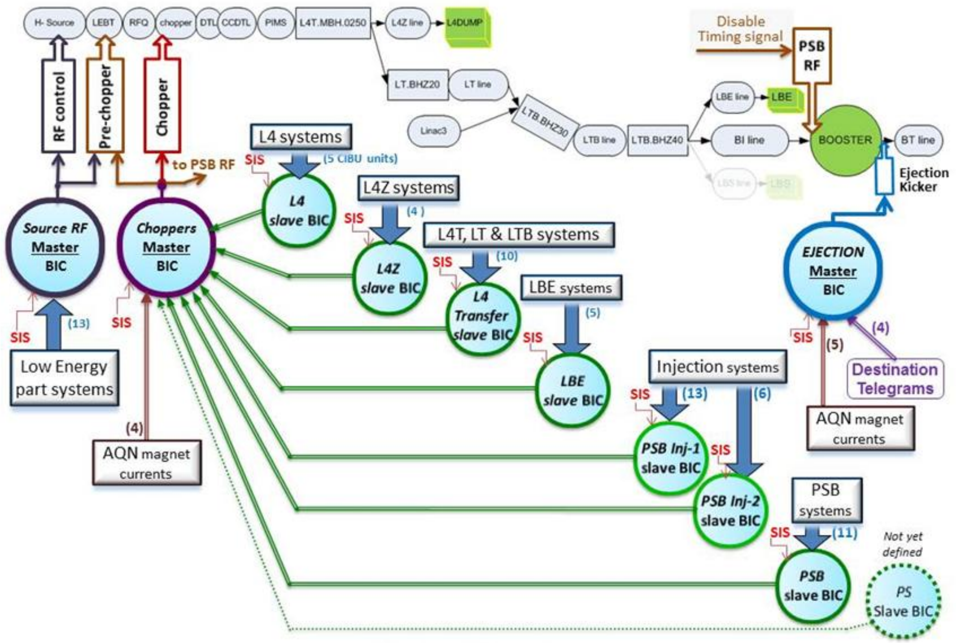
# Architecture

- MPS is comprised of a collection of distributed nodes that collect and process data and send them off the central node.
- The central node collects the data and compares them against a pre-programmed logic table to determine the overall state of the accelerator.
- The output is then distributed to the destination nodes for certain action.



# Truth Table

- Truth Table defines the **relationship** between **inputs** (sensor signals, faults, or triggers) and **outputs** (protective actions, interlocks, or alarms).

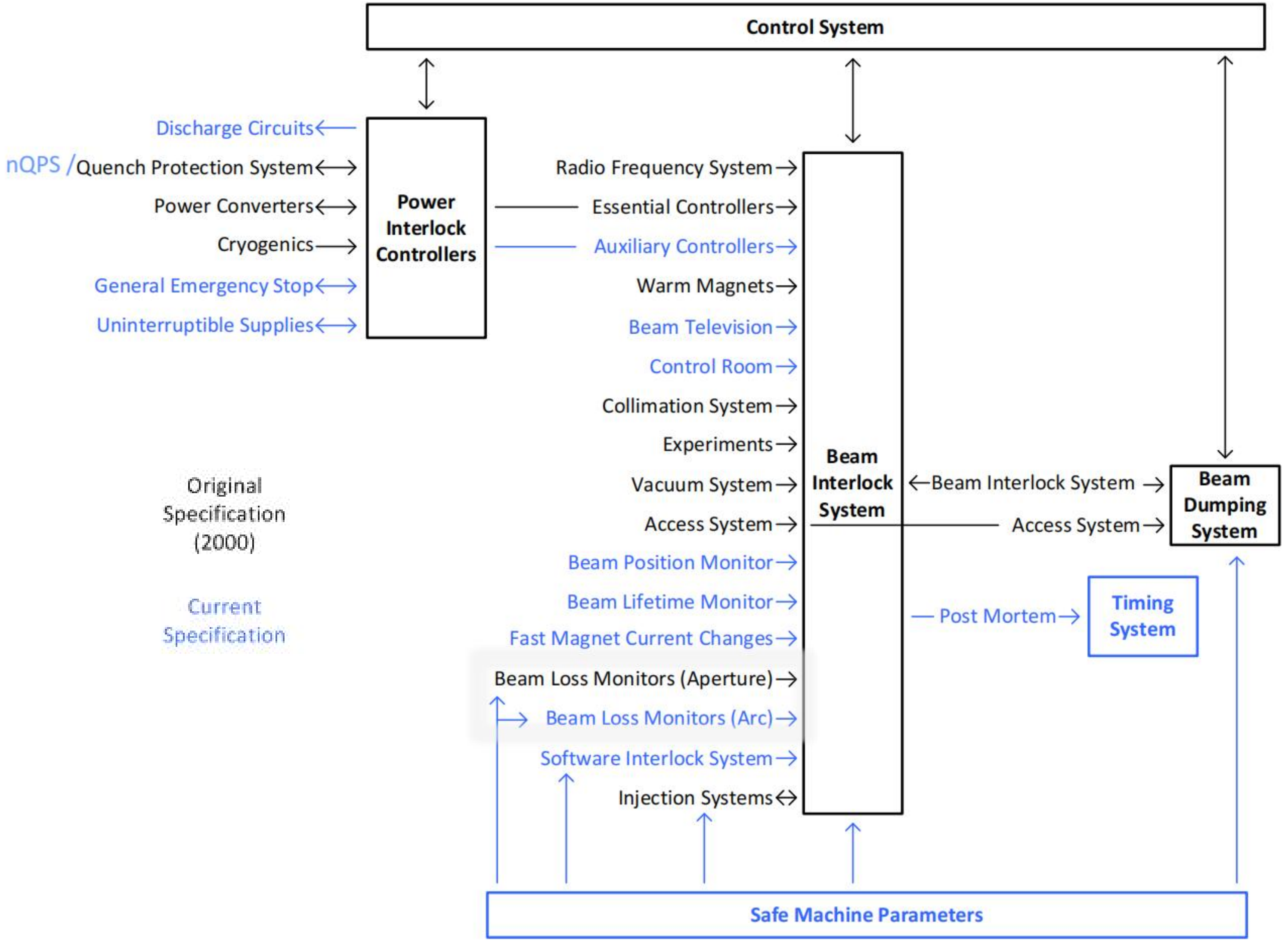


BIS Linac4 & PSB layout

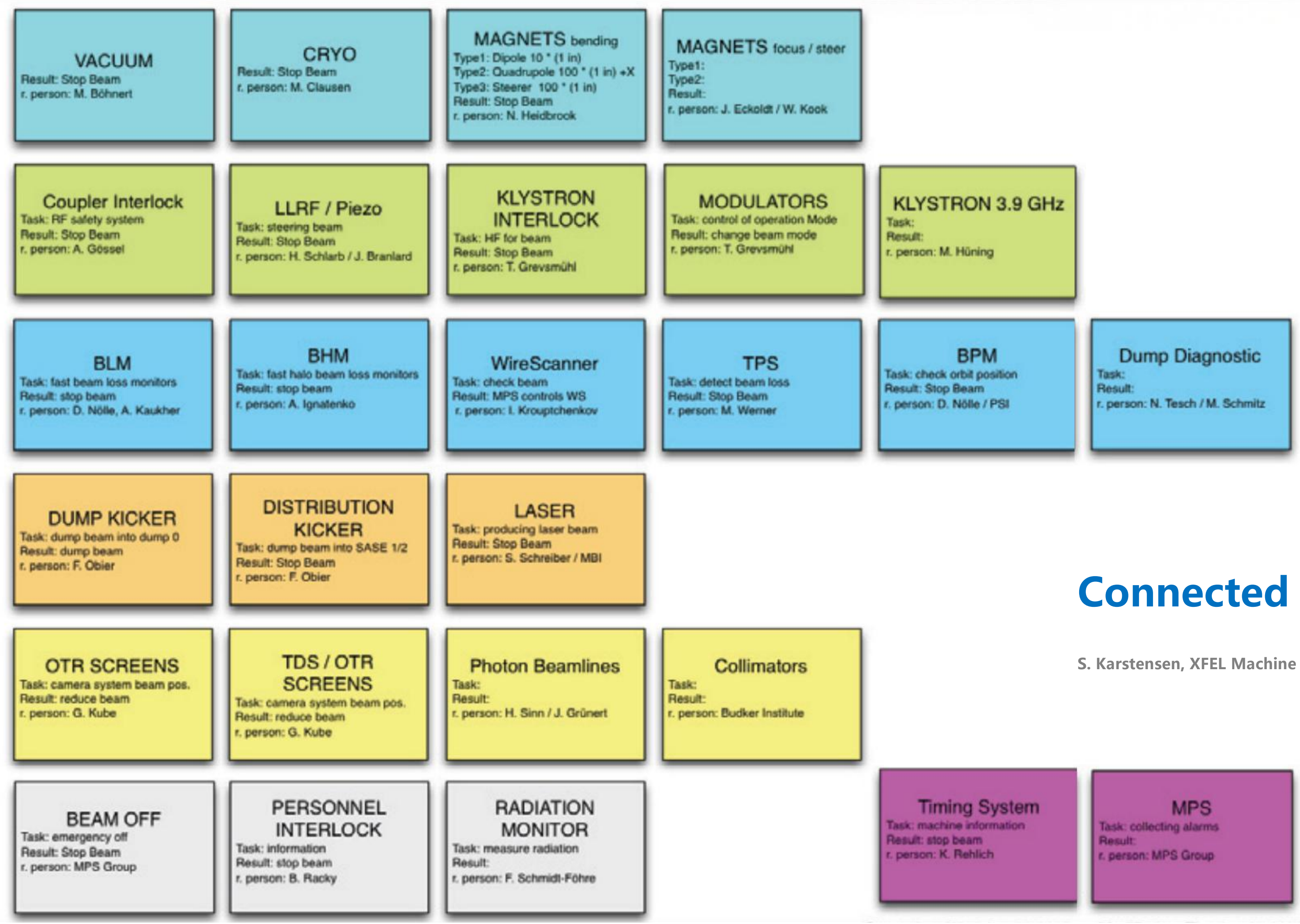
| Ch. | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | OUT | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Interlock Element** | **SIS** | **Linac4 OK** | **AQN L4T.MBH_DUMP** | **L4Z OK** | **AQN L4T.MBH_L4T** | **Linac4 Transfer OK** | **AQN LTB.BHZ40_LBE** | **LBE OK** | **AQN LTB.BHZ40_PSB** | **PSB Injection-1 OK** | **PSB Injection-2 OK** | **PSB OK** | Destination PS | **PS OK** | *Not used* | **Choppers Beam_Permit** | |
| | 1 | 1 | 1 | 1 | 0 | x | x | x | x | x | x | x | x | x | x | 1 | Beam to Dump |
| | 1 | 1 | 0 | x | 1 | 1 | 1 | 1 | 0 | x | x | x | x | x | x | 1 | Beam to LBE |
| | 1 | 1 | 0 | x | 1 | 1 | 0 | x | 1 | 1 | 1 | 1 | x | x | x | 1 | Beam to PSB |
| | 1 | 1 | 0 | x | 1 | 1 | 0 | x | 1 | 1 | 1 | 1 | 1 | 1 | x | 1 | Beam to PS |

# Continue to learn

- MPS architecture is constantly **evolving**, more interlock conditions request an abort of the beam.
- In addition every year some major **changes** to operational systems that require tracking and follow-up (threshold changes, maintenance/ replacement of components, operational tools, procedures,…)



# Interface

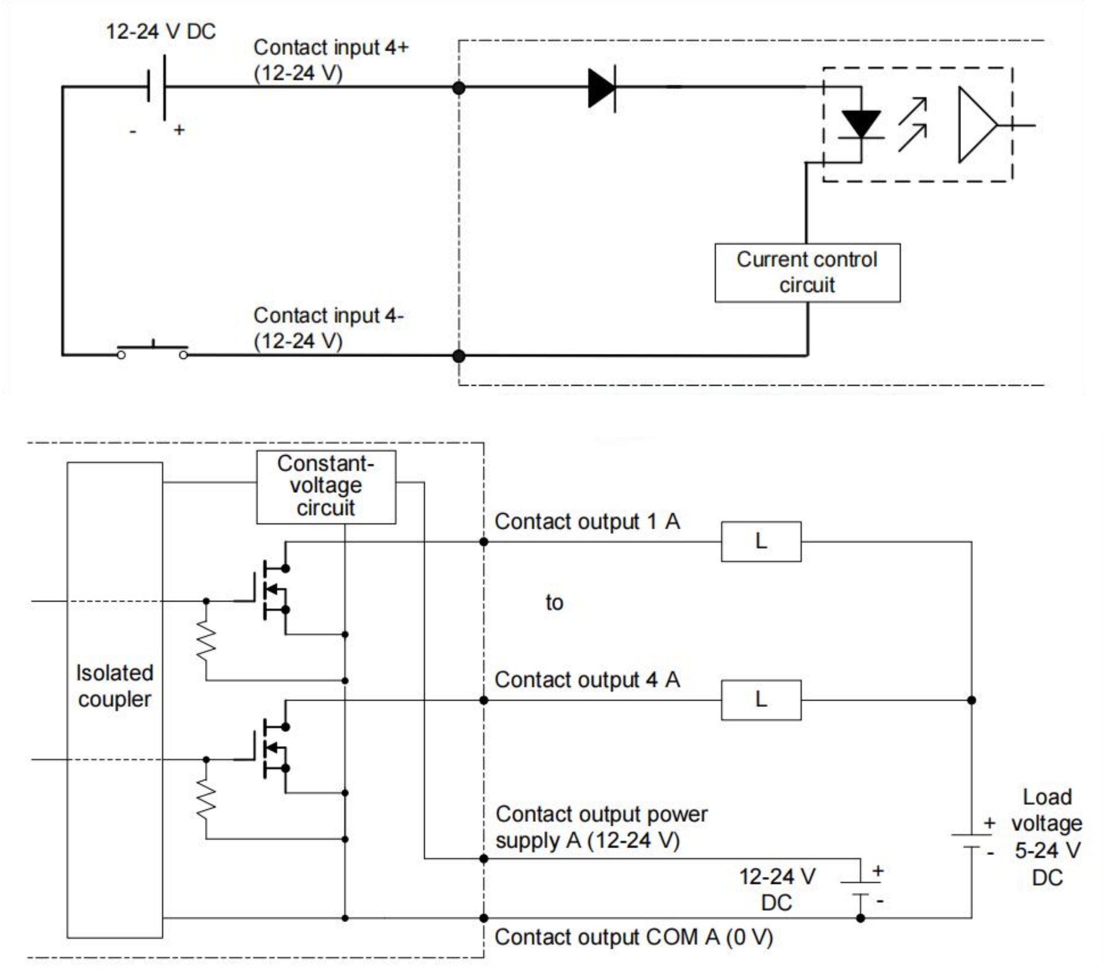



## Connected systems to MPS

S. Karstensen, XFEL Machine Protection System based on uTCA, ICALEPCS 2013

# Hardware

| | PLC | FPGA |
|---|---|---|
| Speed | Moderate (10-50ms) | Ultra-fast (100ns - 10μs) |
| Determinism | Soft real-time (jitter) | Hard real-time (precise timing) |
| Complexity | Simple to medium logic | High-speed parallel logic |
| Flexibility | Easily reprogrammed | Fixed logic (hard to modify) |
| Standards | Certified (SIL 3/PL e) | Custom safety validation needed |
| Development | Faster (ladder logic) | Slower (VHDL/Verilog coding) |

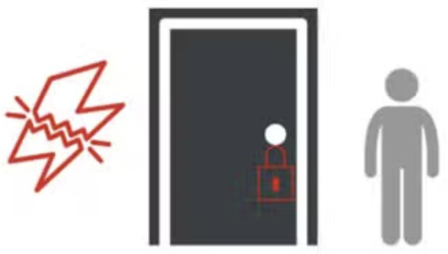



Yokoyama PLC input and output diagram

# Principle

- **Isolation**: Full input/output isolation preferred; output isolation mandatory.
- **Cabling**: Shielded twisted pair (<200m), dual-end grounding (or input-side grounding if restricted).
- **Fail-Safe**: defaults to a safe state in the event of a failure (e.g., power loss, broken wire, or malfunction).

**Fail Safe** - When looking at fail safe locks this means that it's defaut state is actually unlocked.
To keep it locked during normal business operations, power is applied.
Should the power be interupted or fail, the door automaticaly unlocks or releases to let people out of the space.
**That's why it's called "safe"-it's safe forpeople-not the space!**

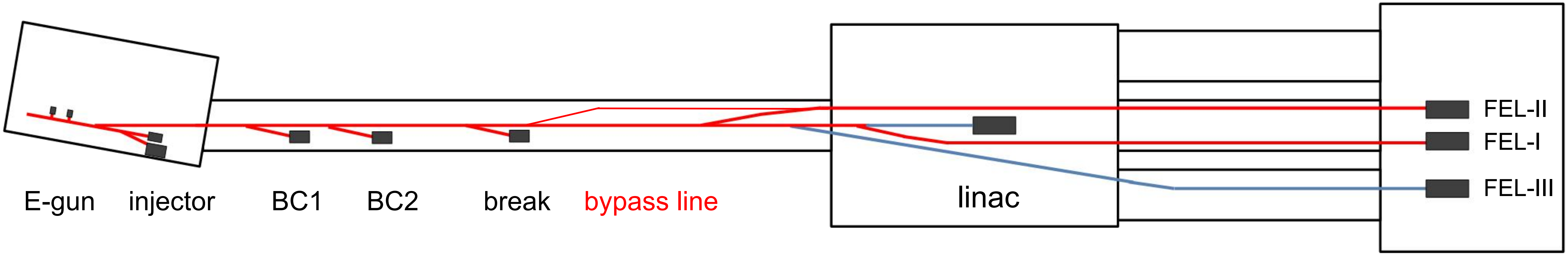


**Fail Secure** - fail secure means that if the power is interrupted, the door stays locked.
That's why it's called "secure"
It's default state is locked or secured. So a falsecure lock locks the door when power is removed.
Often fail secure locks are used for IT rooms or other sensitive areas.

https://www.acslocks.com/fail-safe-vs-fail-secure/

# Operation Mode

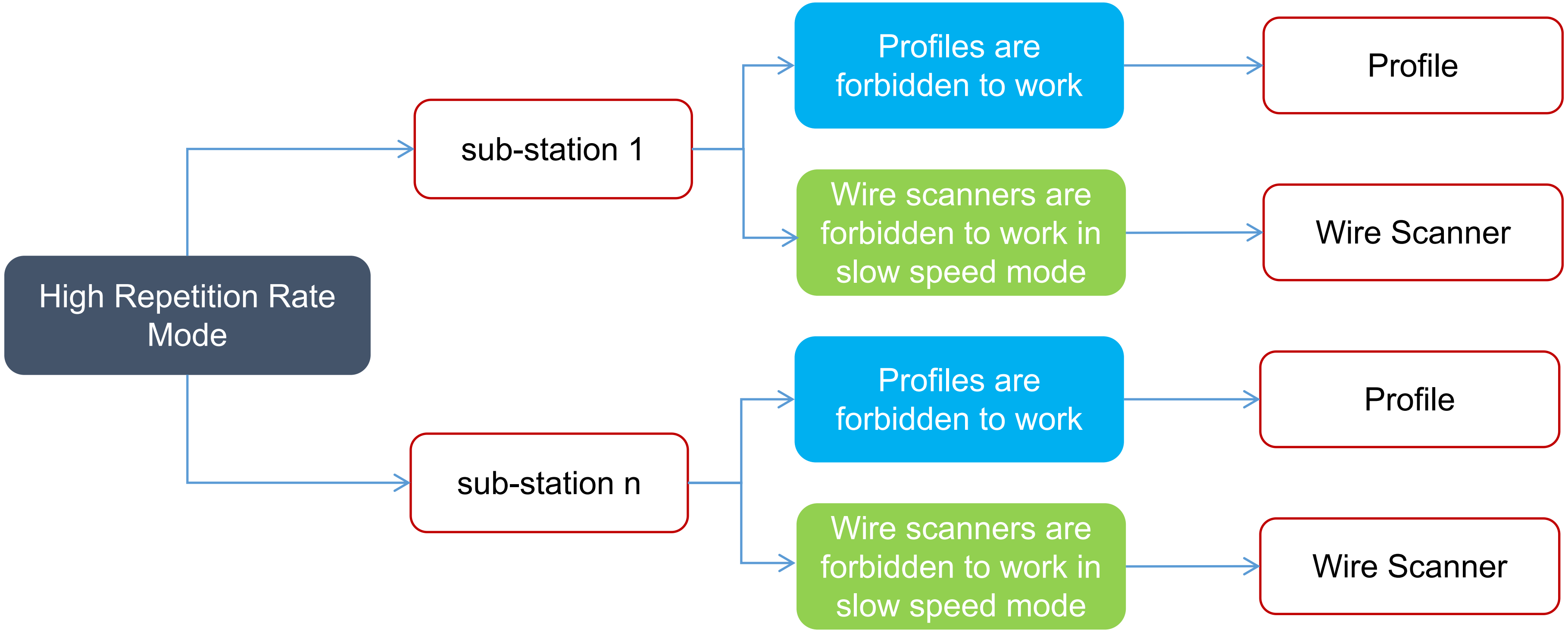
- The Operation Mode is segmented according to the position of the beam dump.
- Different logic can be automatically matched in different modes.




- Injector: E-gun operation mode, injector operation mode.
- Linac:  Bunch Compressor 1, Bunch Compressor 2, Break, Bypass line.
- Undulator and beamlines:  The selection is independent and can be freely combined.
- Low Repetition Rate ≤ 50Hz，Mid Repetition Rate 50 ~ 1kHz，High Repetition Rate ≥1kHz

# Operation Mode

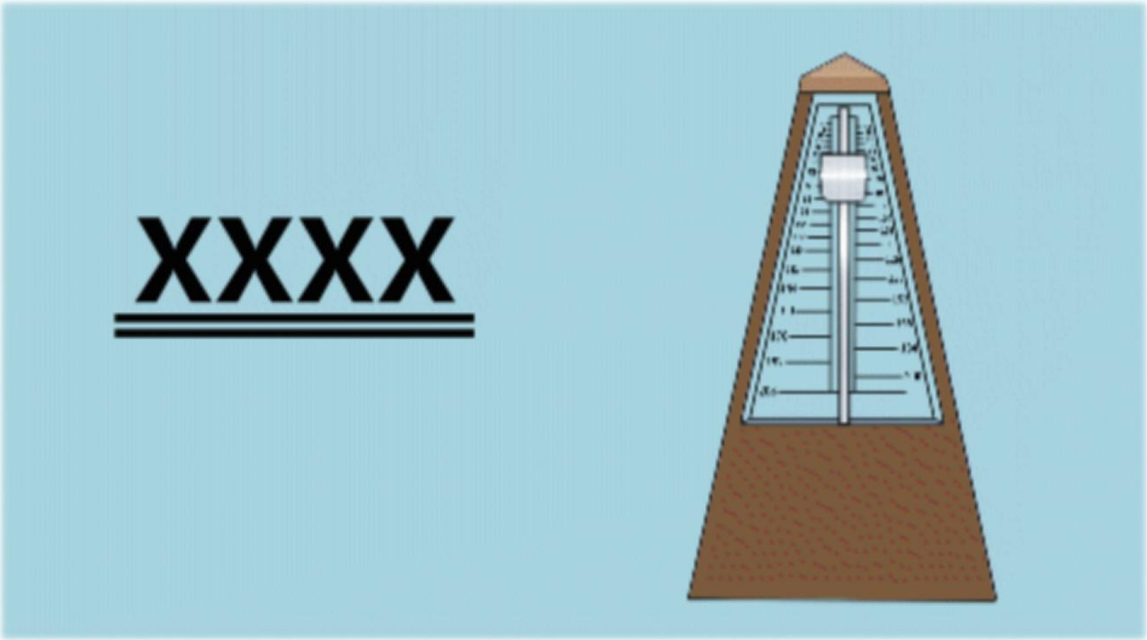
- **High Repetition Rate Mode** (e.g. SHINE)
- The profiles are **forbidden** to work on-line, and the wire scanners are forbidden to work in **slow** speed.



# Timing System

# Timing

- Providing precise **triger** or **reference signals** for lasers, modulators, LLRF, beam & optical diagnostics, ...

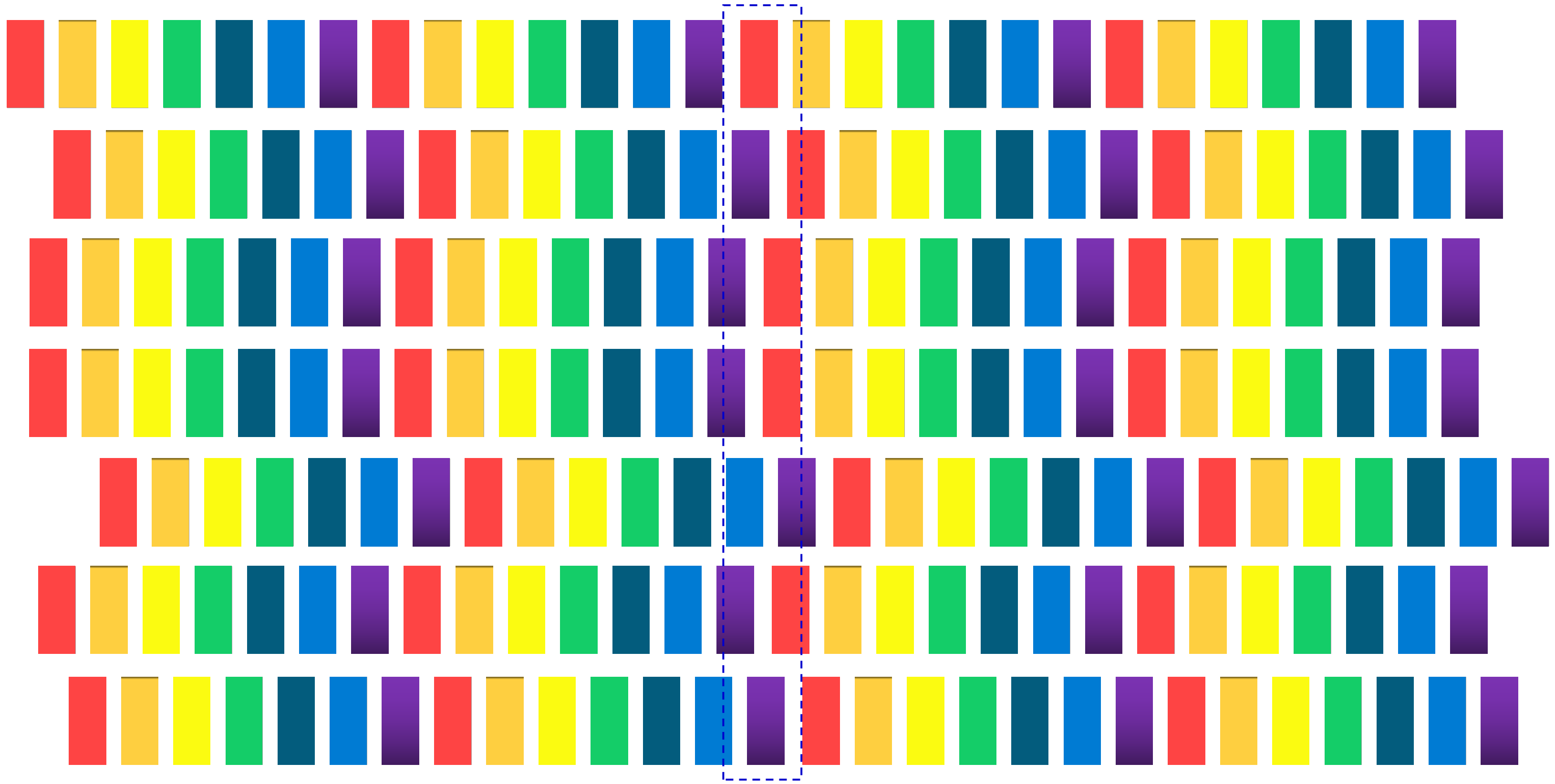



Metronome

Conductor’s baton

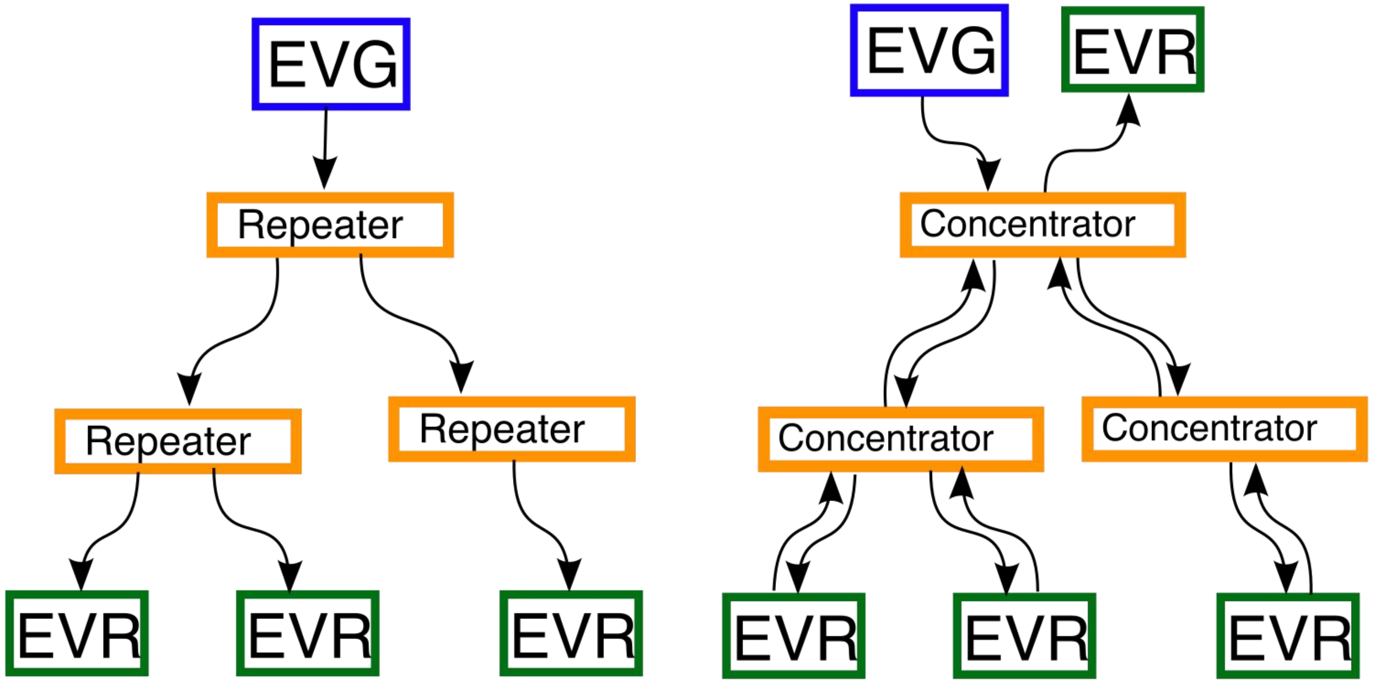
# Timing

- Providing precise **triger** or **reference signals** for lasers, modulators, LLRF, beam & optical diagnostics, ...
- Providing hardware **BunchID / PulseID** for beam parameter analysis and failure diagnosis.



# MRF (1990s ~ )

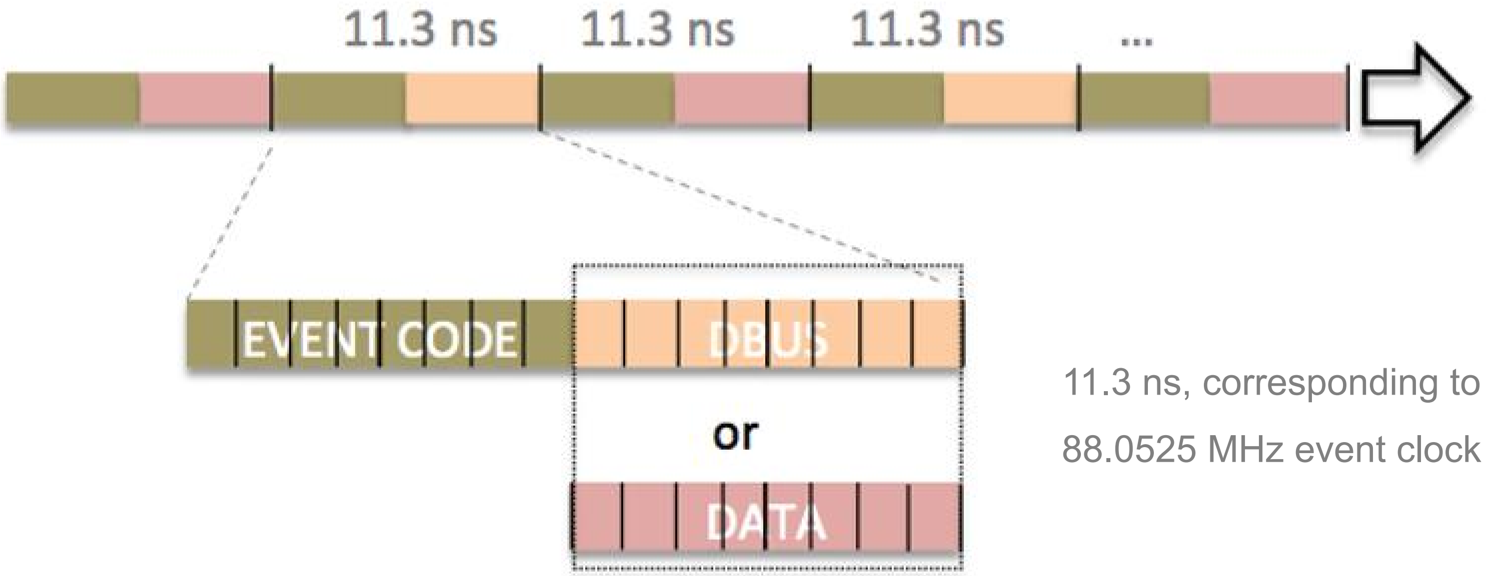
- MRF (**Micro-Research Finland**) Timing System is a high-precision **event synchronization system** widely used in particle accelerators.
- It is capable of generating subharmonic frequency signals, triggers and sequences of events, etc.
- A basic setup of the timing system consists of an Event Generator (**EVG**), the distribution layer (Fan-Out, or Repeater/Concentrator) and Event Receivers (**EVR**).




https://mrftiming.readthedocs.io/en/latest/

# Event Stream

- The event stream protocol is based on 8b10b encoded characters.
- Each frame of the stream consists of two bytes.
  - The first byte is dedicated for transmitting timing events, and always contains an event code.
  - The second byte can be configured as distributed bus bits or synchronous data transmission.

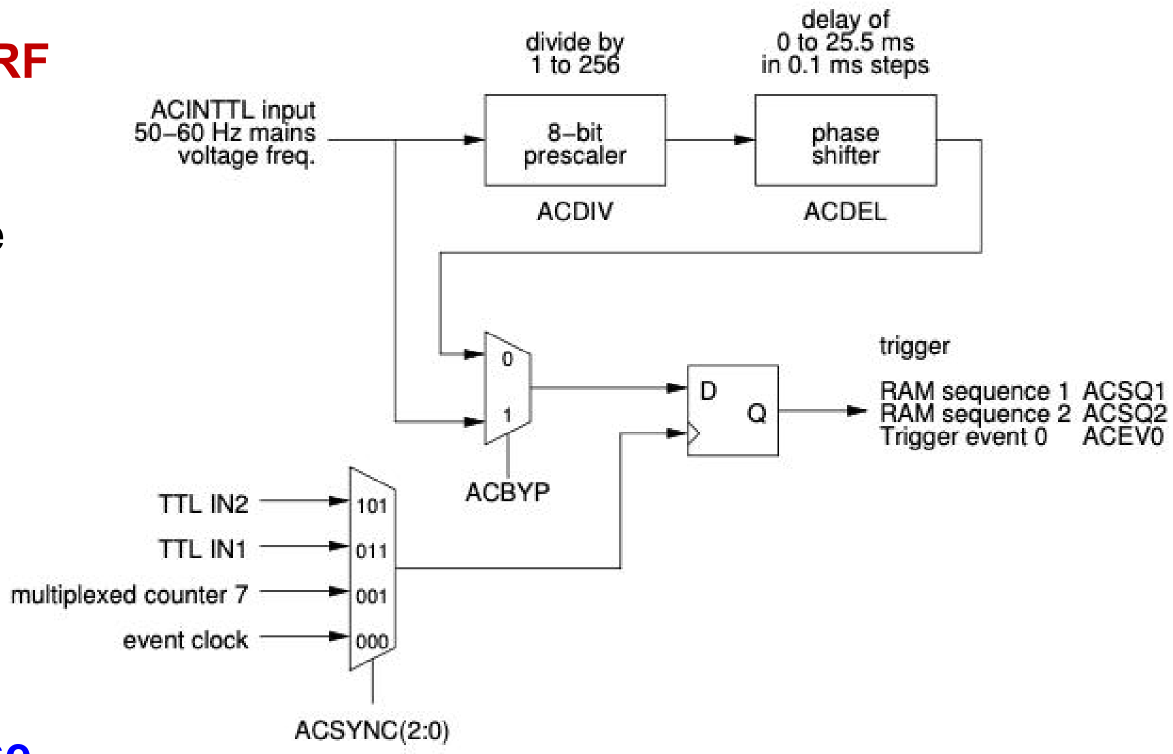



11.3 ns, corresponding to 88.0525 MHz event clock

https://mrftiming.readthedocs.io/en/latest/

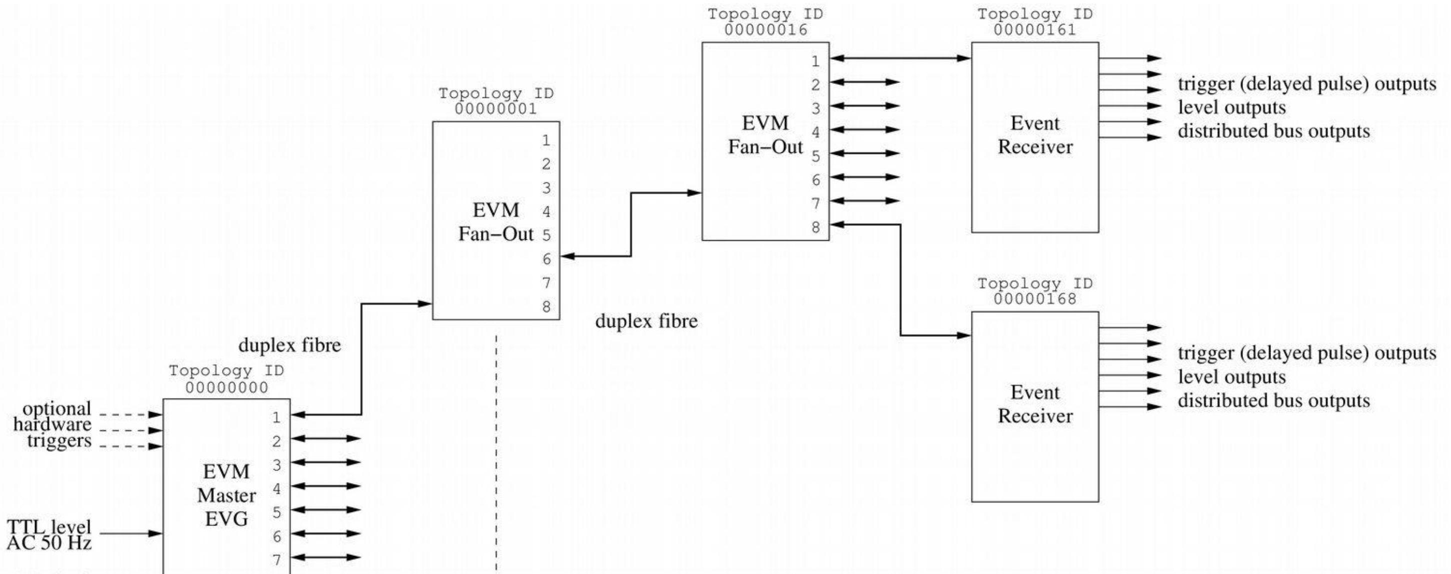
# Event Generator (EVG)

- **Event Generator** generates the event stream and sends it out to an array of **Event Receivers**.
- Events are sent out by the event generator as **event frames (words)** which consist of an eight bit event code and an eight bit distributed bus data byte.
- The event transfer rate is **derived** from an **external RF clock** or optionally an on-board clock generator.
- The Event Generator provides synchronization to the **mains voltage frequency** or another external clock.
- The mains voltage frequency can be divided by an eight bit programmable divider. The output of the divider may be delayed by 0 to 25.5 ms by a phase shifter in 0.1 ms steps to be able to **adjust the triggering position relative to mains voltage phase.**



# Event Generator (EVG)

- With **different cable lengths**, long distances and **thermal gradients**, the propagation delays could **drift** and disturb operation in cases where long-term timing **stability** is critical.
- The **delay compensation** can be used to stabilize the system against e.g, thermal drifts of optical cables.
- In the **300-series** event system, an active delay compensation feature was added.

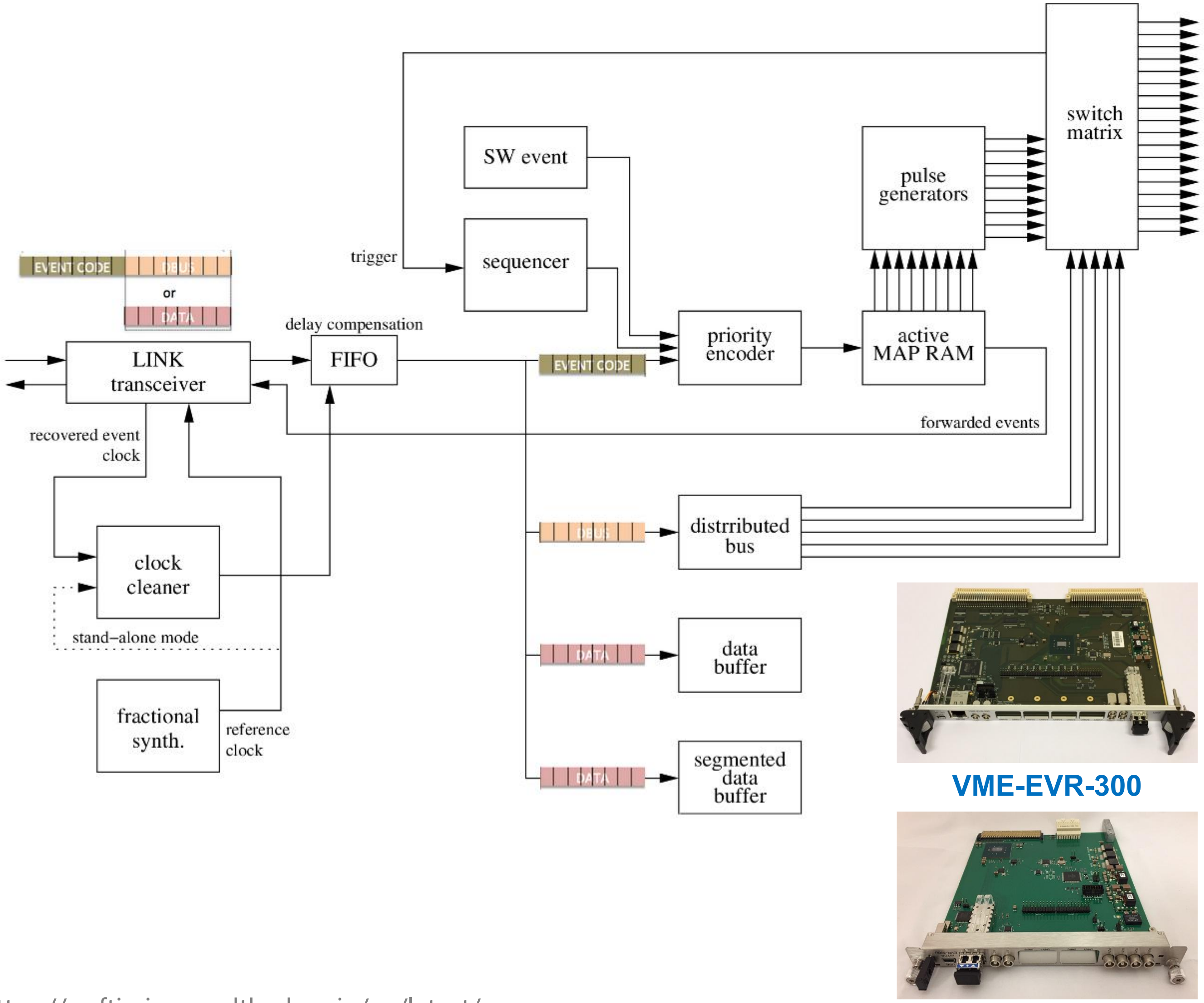



Event Master (EVM) can be configured as Event Generator or Fan-Out

**VME-EVM-300**

# Event Receiver (EVR)

- EVR **decode** timing events and signals from an optical **event stream** transmitted by EVG.
- EVR lock to the phase event clock of the Event Generator and are thus phase **locked to the RF reference**.
- EVR convert event codes to hardware outputs.
- It also generate software interrupts and store the event codes with globally distributed timestamps into FIFO memory to be read by a CPU.

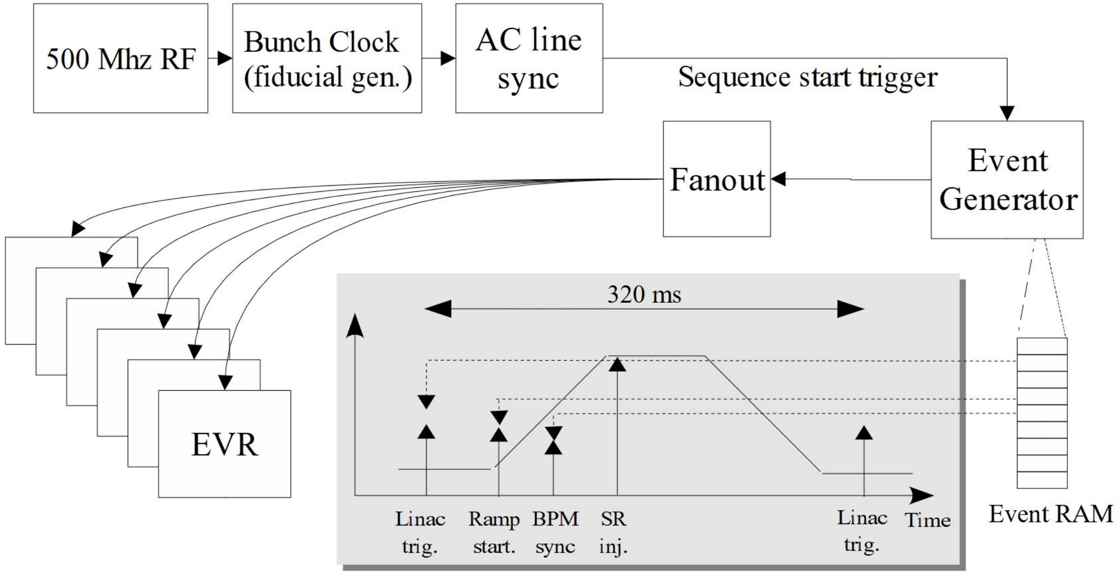


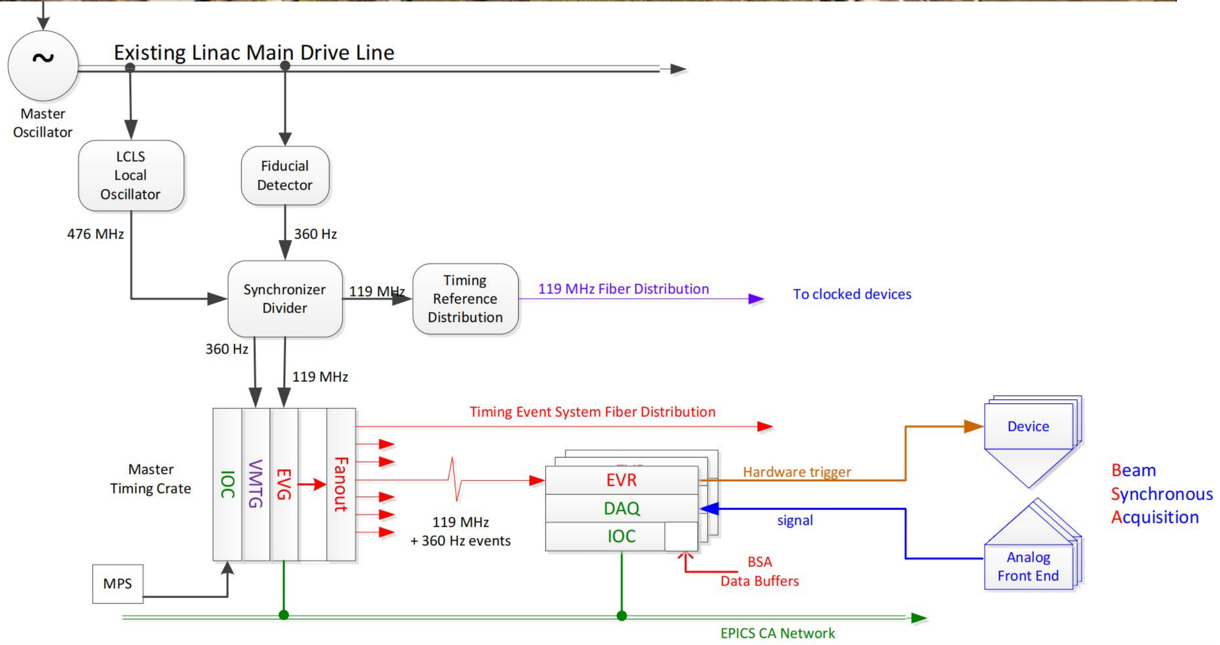

VME-EVR-300

mTCA-EVR-300U

https://mrftiming.readthedocs.io/en/latest/

# SLS and LCLS-II Timing System



# White Rabbit

- The **White Rabbit Project** is a multilaboratory, multicompany and multinational **collaboration** to develop new technology that provides a versatile solution for control and data acquisition systems.
- The project was started within an effort to renovate the current CERN control and timing system.
- The main features of the White Rabbit Network are:
  - **sub-ns** accuracy and **ps** precision of synchronization
  - connecting thousands of nodes
  - typical distances of **10 km** between network elements
  - Gigabit rate of data transfer
  - **fully open** hardware, firmware and software
  - commercial availability from many vendors

**02/2008 : Project start**

**12/2010 : Basic Ethernet switching demonstrated**

**04/2011 : PTP working on a WR node**

**05/2012 : The first system was successfully in CERN**

**...**

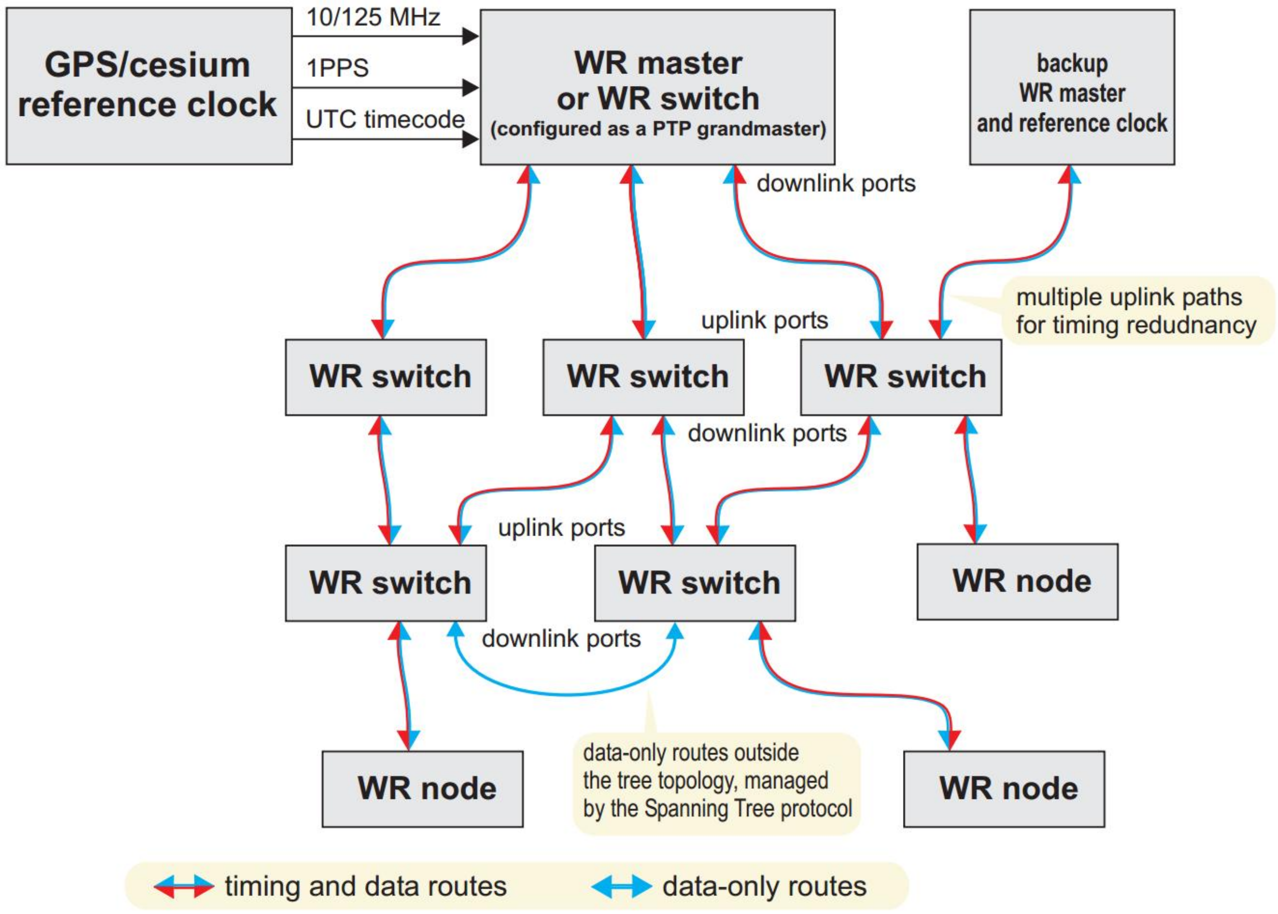
**06/2020 : IEEE-1588-2019 v2.1 officially published**

**...**

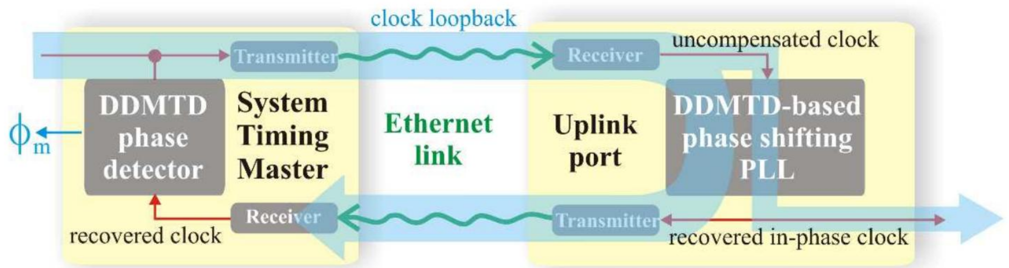
- CERN, GSI, ESRF, ELI-ALPS, ELI-BEAMS, Fermilab, ...
- LHAASO, SHINE, SXFEL, HIAF, CiADS, ...

https://white-rabbit.web.cern.ch/

# White Rabbit




- ❑ Synchronous Ethernet (**SyncE**)
  - frequency synchronization over Ethernet
- ❑ Precision Time Protocol (**IEEE 1588**)
- ❑ Double Dual Mixer Time Difference (**DDMTD**)

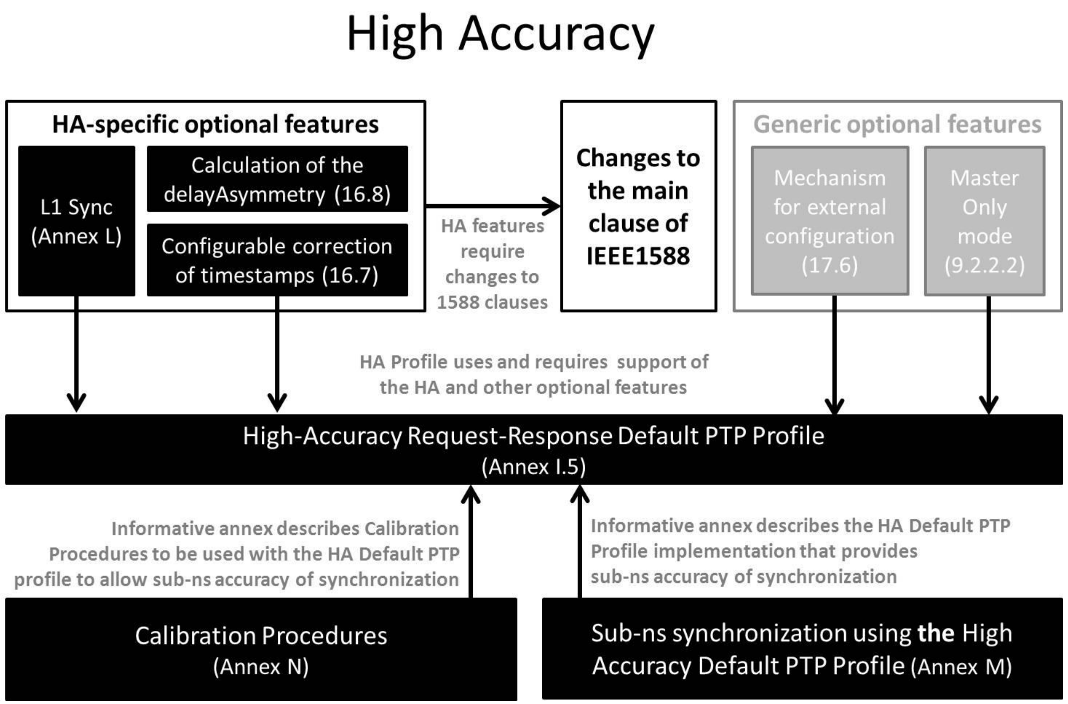


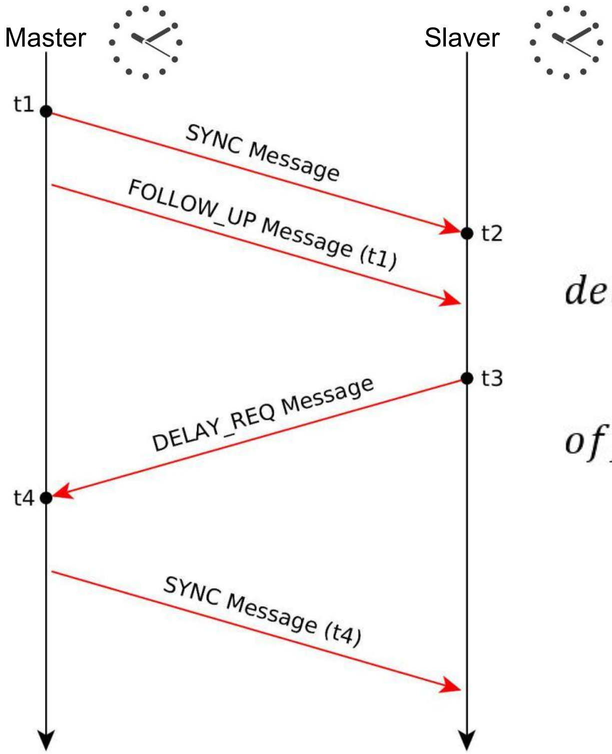

measure ultra-small time intervals or phase differences between two signals with picosecond-level resolution

https://white-rabbit.web.cern.ch/

# PTP

- The **Precision Time Protocol** (**PTP**) is specified in the IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems.
- The currently active **2019 edition** was preceeded by 2008 and 2002 editions.



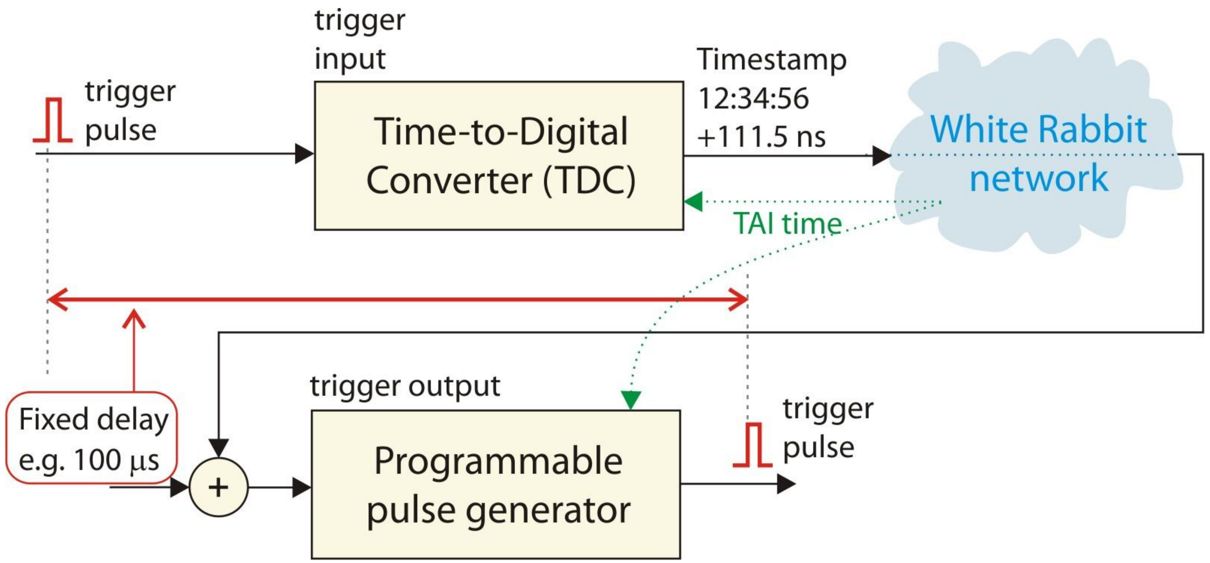



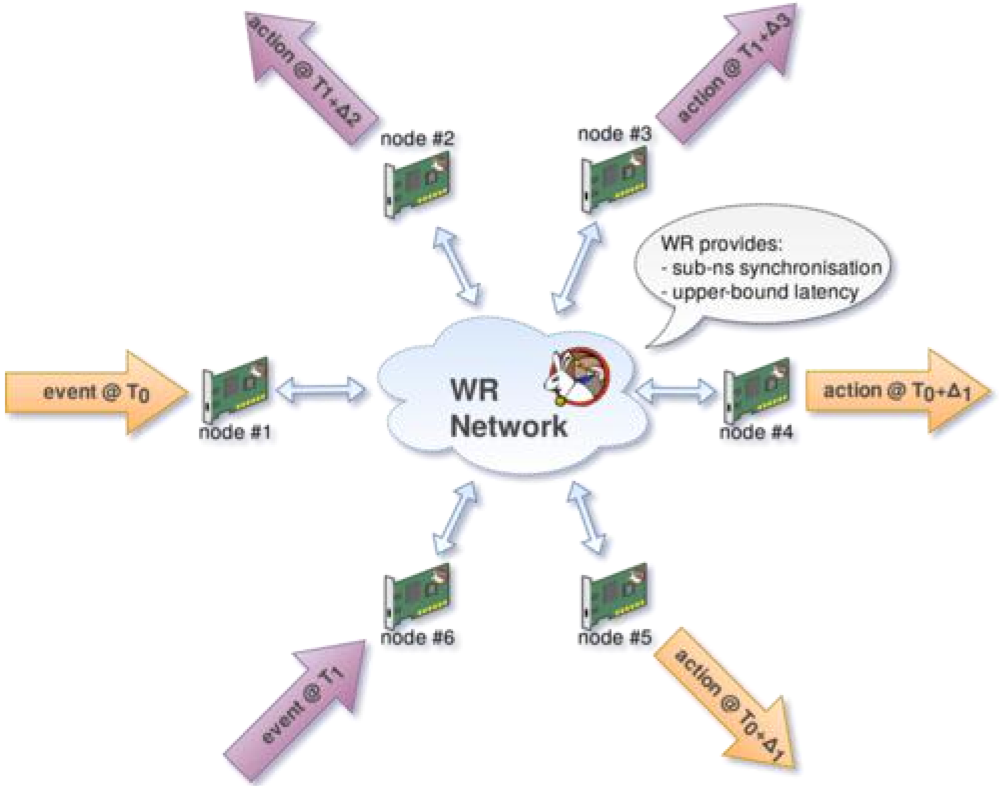

$$delay = \frac{(t_2 - t_1) + (t_4 - t_3)}{2}$$

$$offset = (t_2 - t_1) - delay$$

# WRTD

- **White Rabbit Trigger Distribution** (**WRTD**) is a generic framework for distributing **triggers** (events) between Nodes over a White Rabbit network.

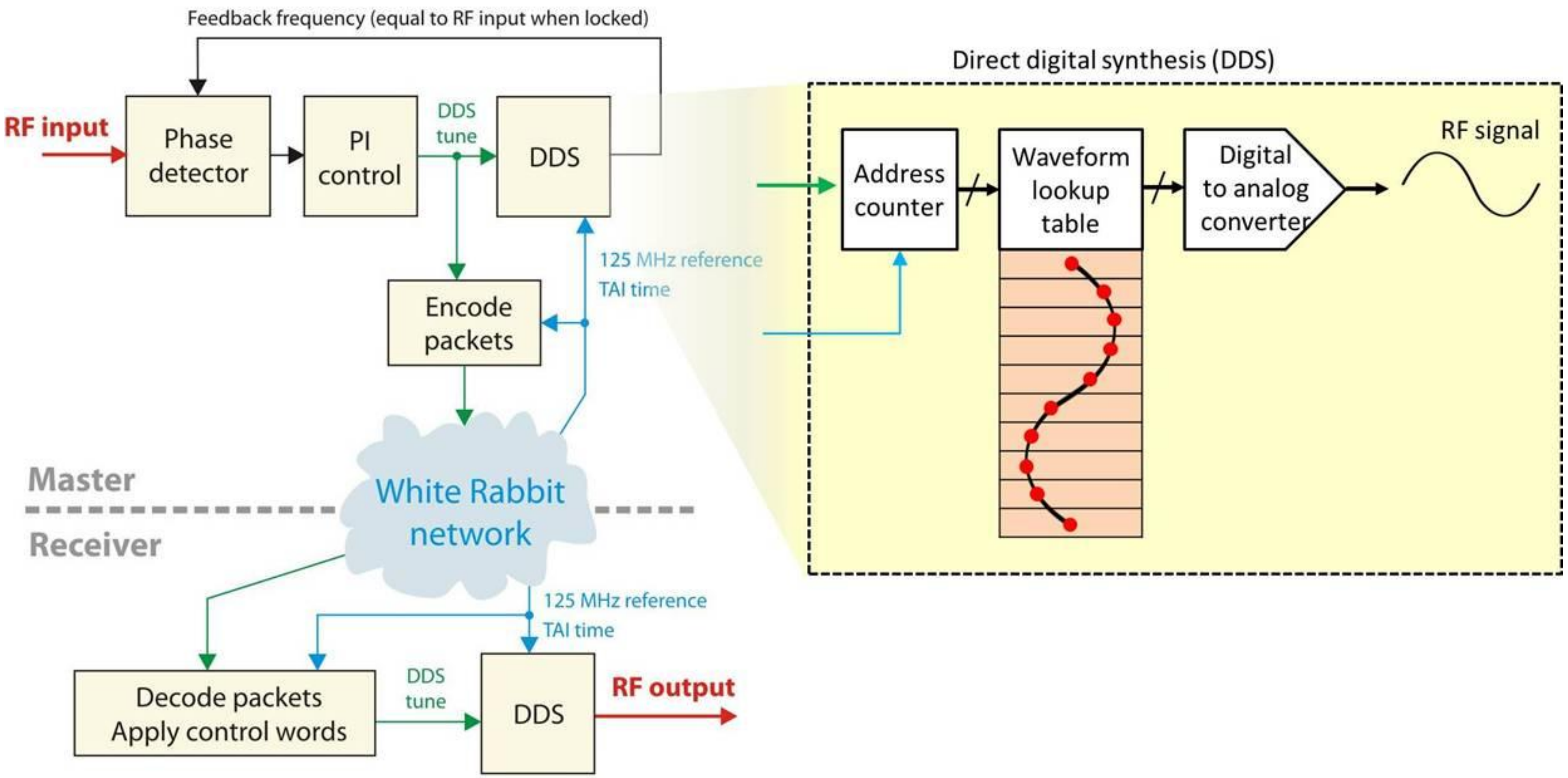



https://www.white-rabbit.tech/

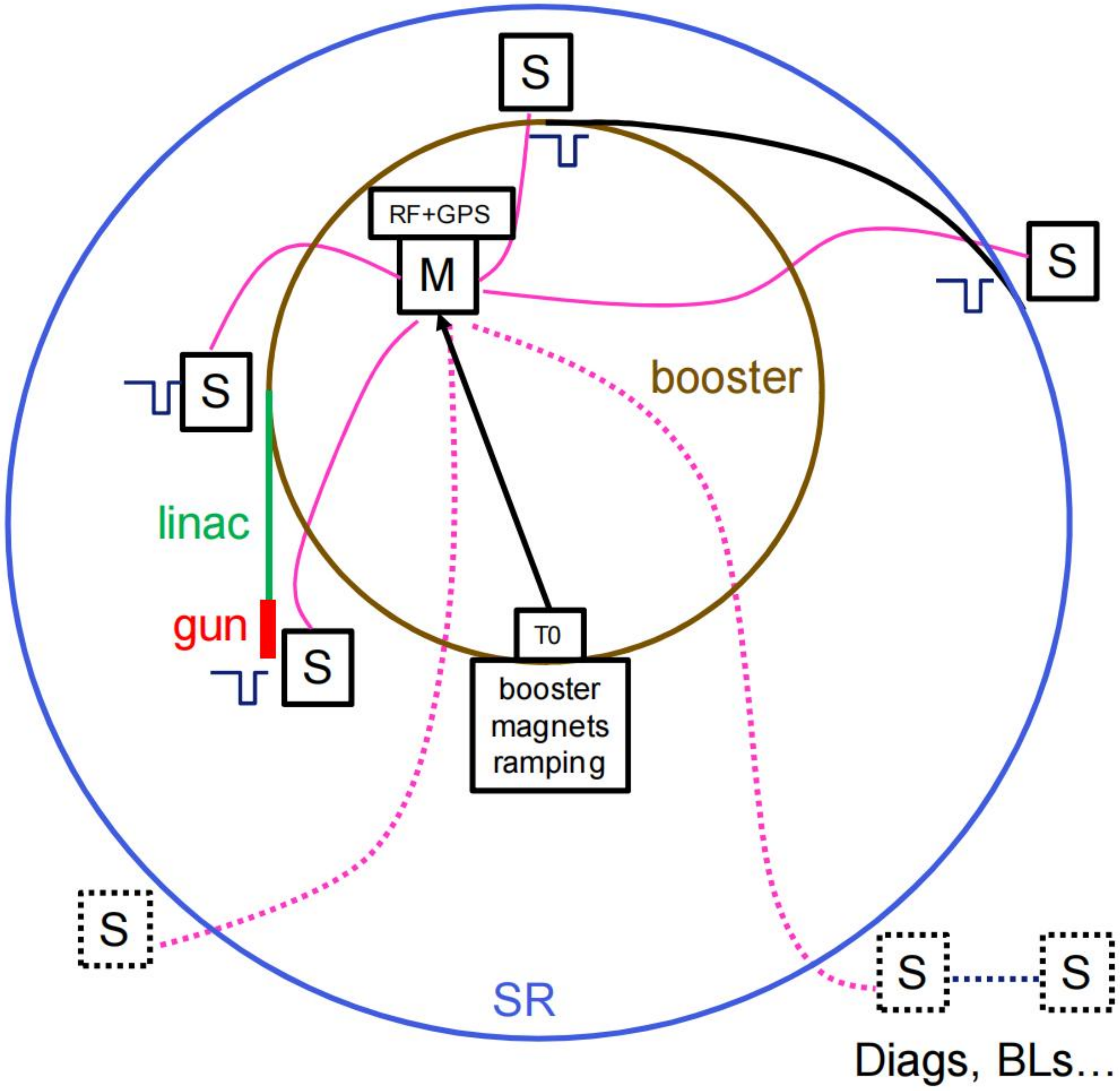
# RF over WR

- RF over WR allows real-time transmission of **frequency tuning words** over a White Rabbit network.
- WR is used to distribute the base phase-aligned WR clock signal and the frequency tuning words representing the RF signals.




https://www.white-rabbit.tech/

# ESRF Timing system




**optical fibre network**

1 master (CTRM)

+ satellite slaves

(WR switches not shown)

**MASTER** manages

RFoE

booster_ramp_start (T0)

**SLAVES** manage

time (RF+UTC)

time stamping of inputs

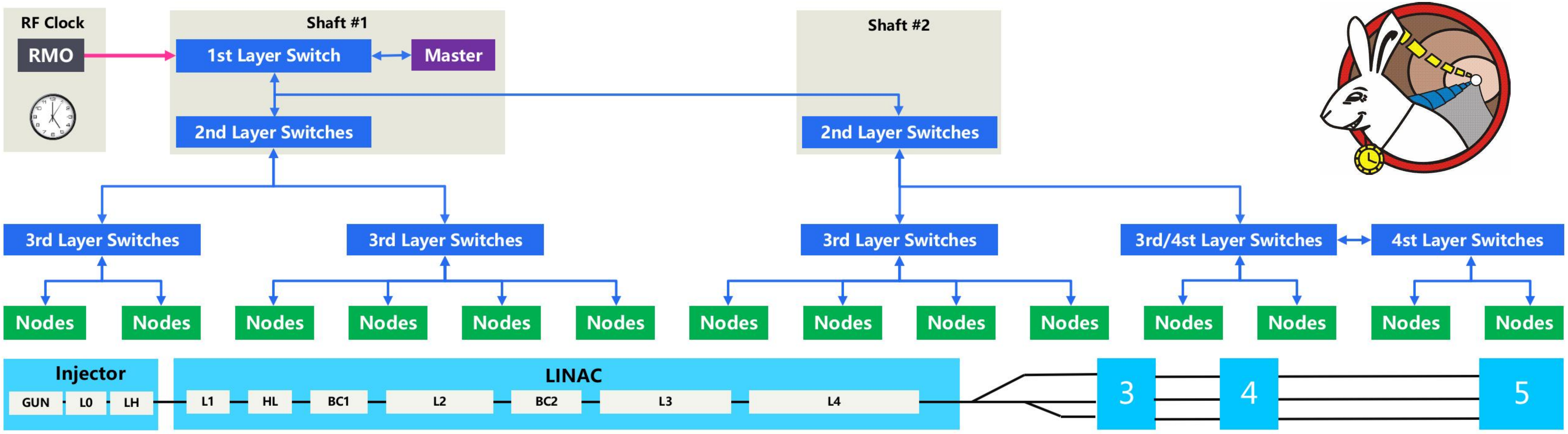
main sequencer

local pulse production

G.Goujon, tucpl01_talk, ICALEPCS 2017

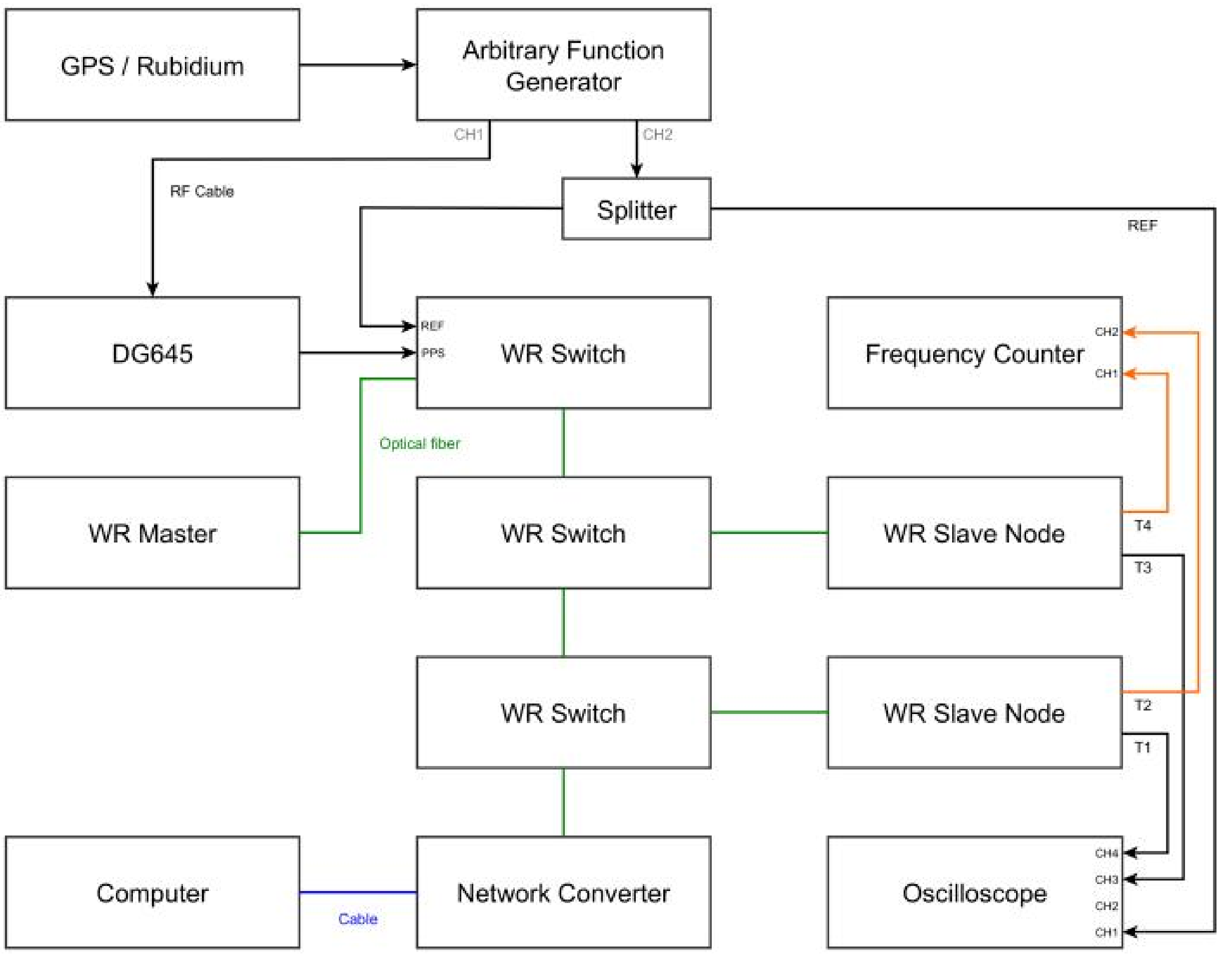
# SHINE Timing System

- Beam-synchronous **trigger** signal distribution (1.003086MHz timing signals over ~ 3.1 km).
- Random-event **trigger** signal distribution (event signals, such as beam loss, machine snapshot, etc.).
- **Bunch ID,** which provides a data alignment basis for beam parameter analysis and failure diagnosis.




**non-standard clock** White Rabbit timing system

# SHINE Timing System




**Beam-synchronous trigger signal**

- Jitter between the slave node output and reference signal < **10ps**
- Jitter between slave nodes outputs < **5ps**

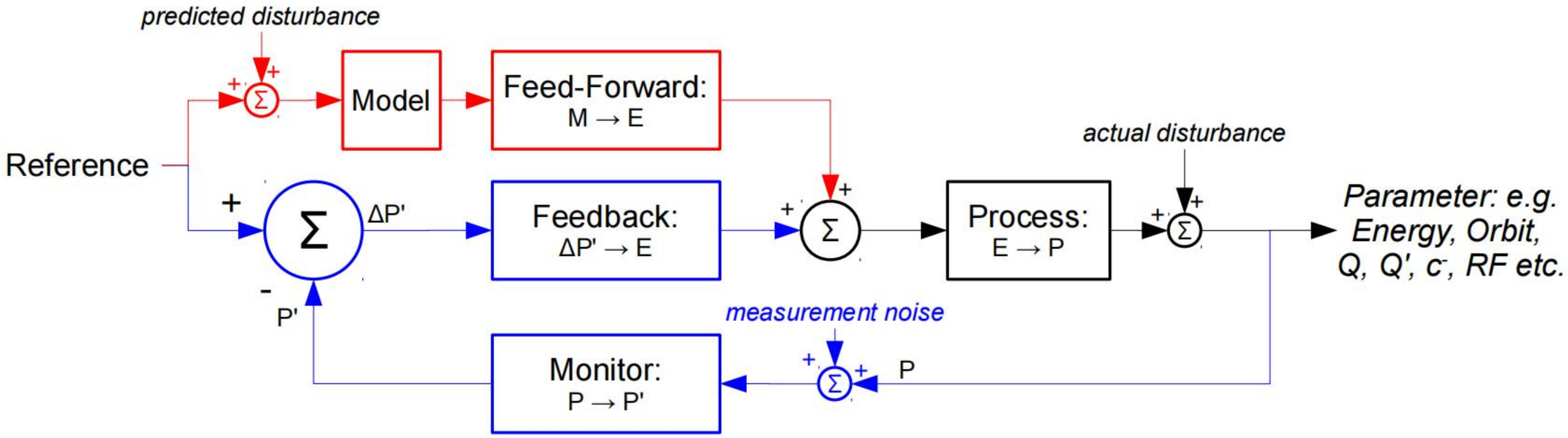
**Clock Phase Noise**

- 10Hz - 10MHz jitter < **2ps**
- Agilent E5052B Signal Source Analyzer

# Feedforward & Feedback

# Feedforward & Feedback

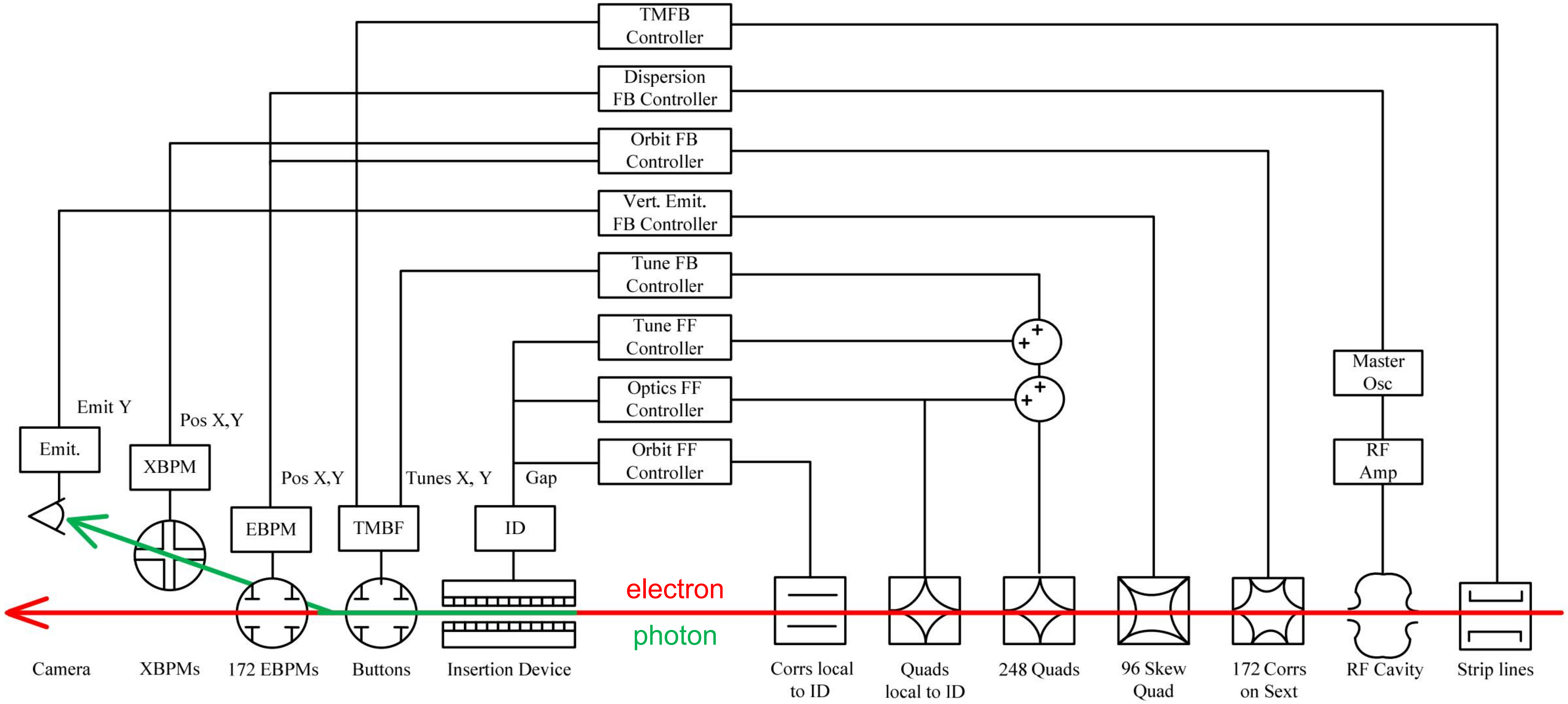
- **Feed-Forward (FF)**
  - Steer parameter using precise process model and disturbance prediction.
- **Feedback (FB)**
  - Steering using rough process model and measurement of parameter.




Ralph J. Steinhagen, Feedback Control for Particle Accelerators, PCaPAC 2016

# Feedforward & Feedback

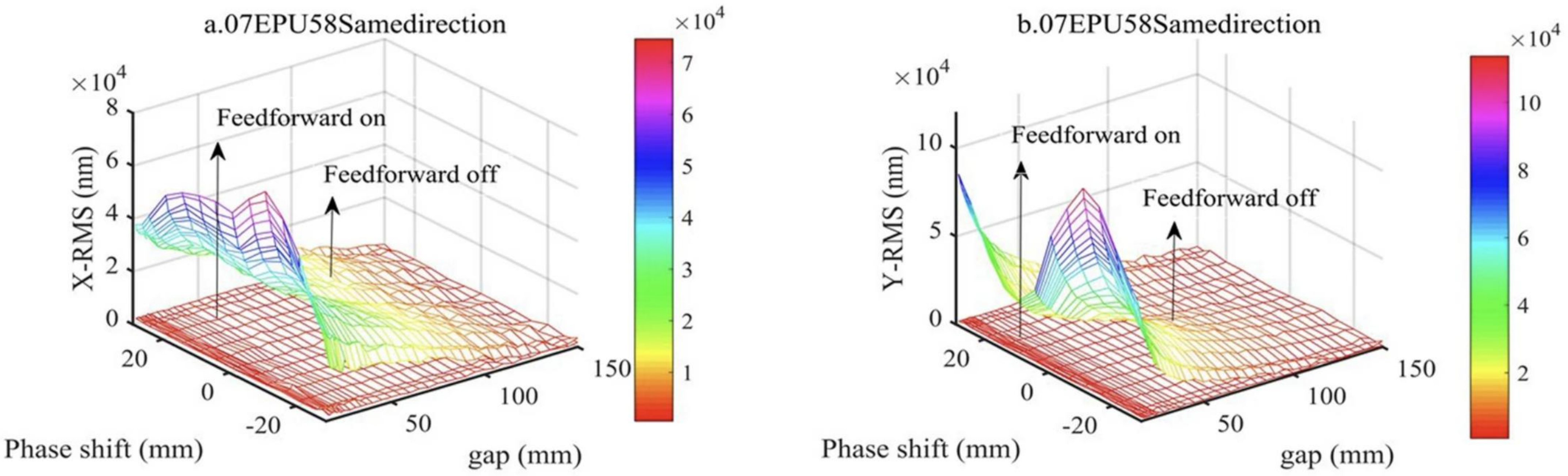
## Conceptual view of the feedback (FB) and feed-forward (FF) correction schemes on the Diamond storage ring




M.T. Heron, TUPRI081, IPAC2014

# Feedforward of IDs effects at SSRF

- Insertion Devices (IDs) cause closed-orbit distortions, tune drift, and coupling distortions in the storage ring.
- **ID orbit feedforward compensation system** based on a response matrix using corrector coils.
- ID_IOC (gap and phase shift) ➡ FF_IOC (Python based) ➡ PS_IOC (current of the corrector coils)




**Changes in the same-direction orbit distortion with gap size and phase shift before and after feedforward compensation of the 07EPU58**

Xin-Zhong Liu, Shun-Qiang Tian, etc. Nuclear Science and Techniques, Volume 33, article number 70, (2022)

# FB Design Paradigms – Stability

**Perturbation Sources or "Know your enemy"**

- **Environmental sources:**

  (mostly propagated through quadrupoles/girders)
  - temperature and pressure changes
  - ground motion, tides
  - 'cultural noise'
- **Machine inherent sources:**
  - decay and snap-back of magnetic multipoles,
  - cooling liquid flow, pumps/ventilation vibrations
  - eddy currents
  - changes of machine optics (feed-down effects)
  - machine impedance, trapped RF modes/wake-fields
  - Intensity-related and collective effects
- **Machine element failures:**
  - magnet quenches, power converter/RF trips, ...
  - corrector circuits

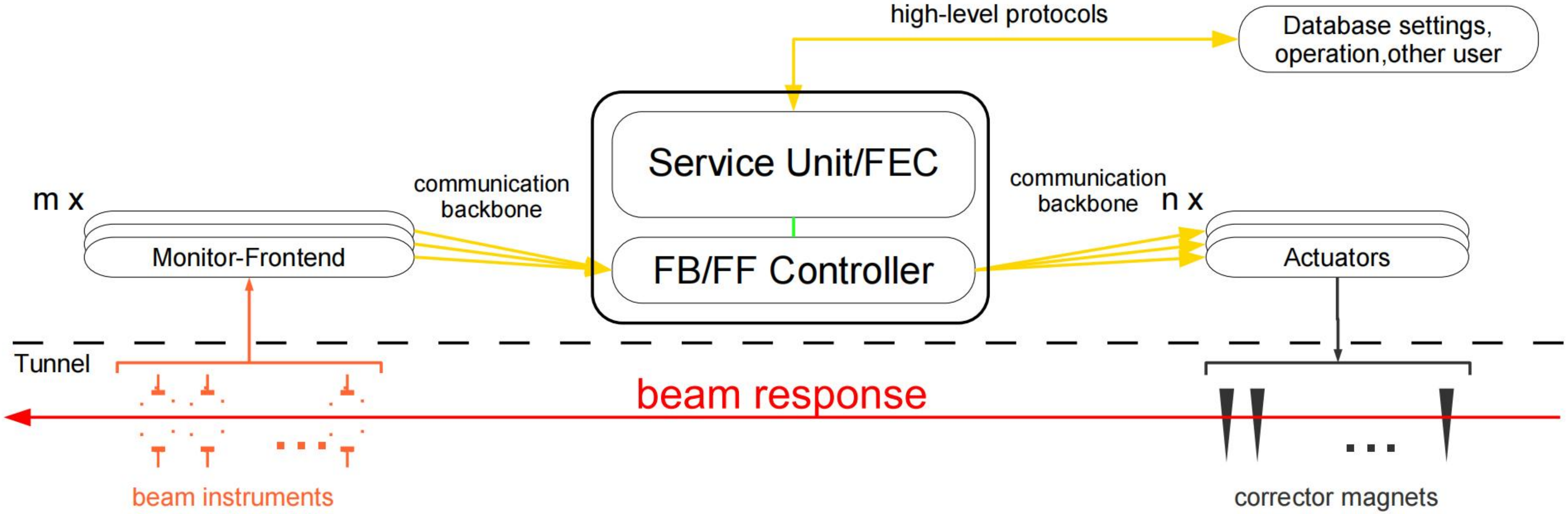
months
↓
weeks
↓
days
↓
hours
↓
sec
↓
ms
↓
minutes
↓
us
↓
ns

Stability defined by accelerator design

Stability defined by feedback design

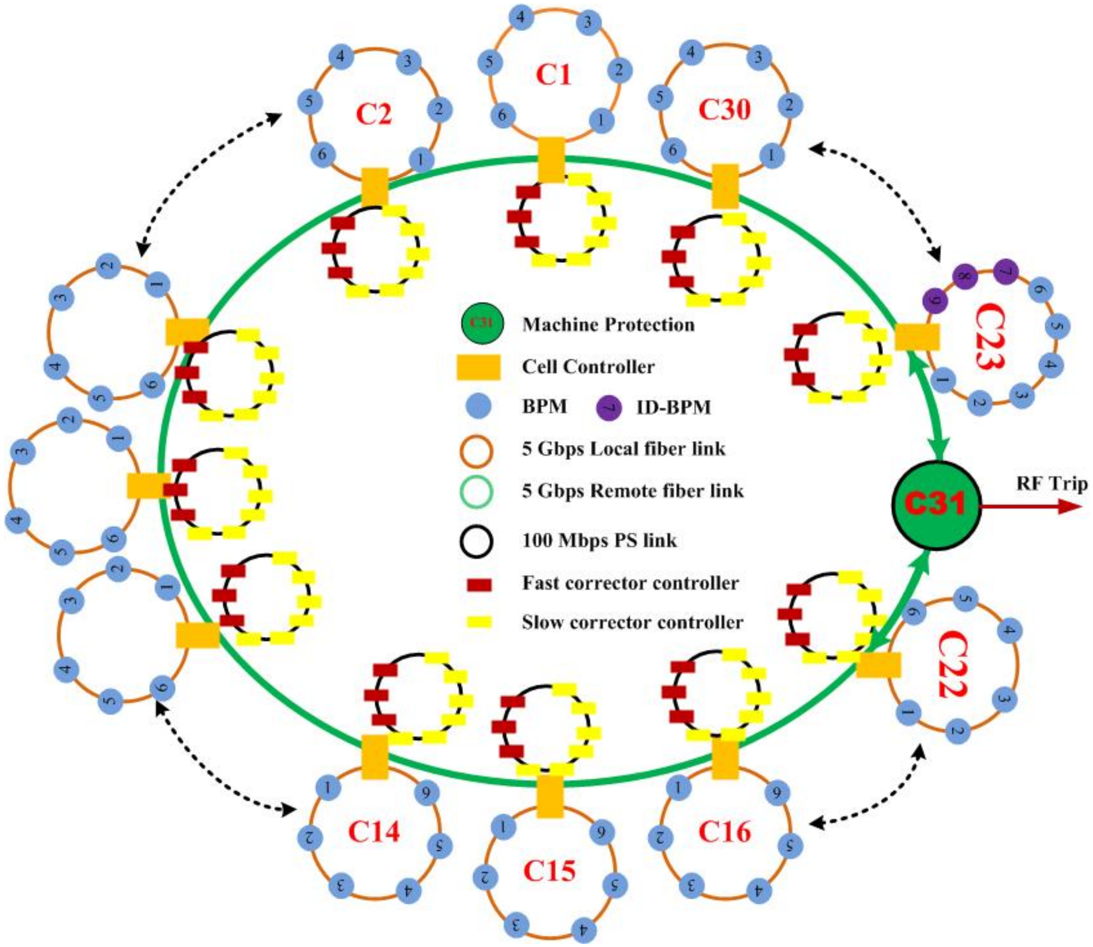
# Feedbacks - Control Layout & Implementation

- fairly generic, typically MIMO & often split into two sub-systems.
  - **Feedback Controller**: actual feed-forward/feedback controller logic.
    - specific implementation depends on the bandwidth requirement.
  - **Service Unit**: Interface to control system.
- Overall strength depends on the knowledge / reliability of the weakest link in the chain.

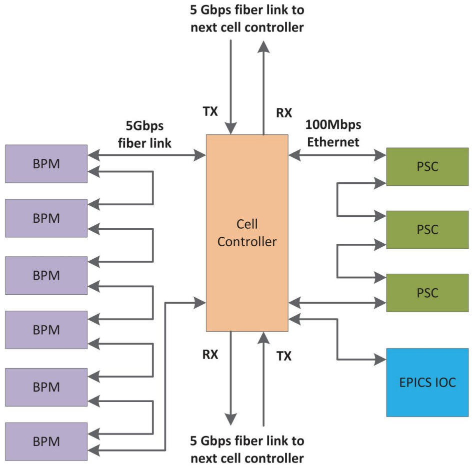


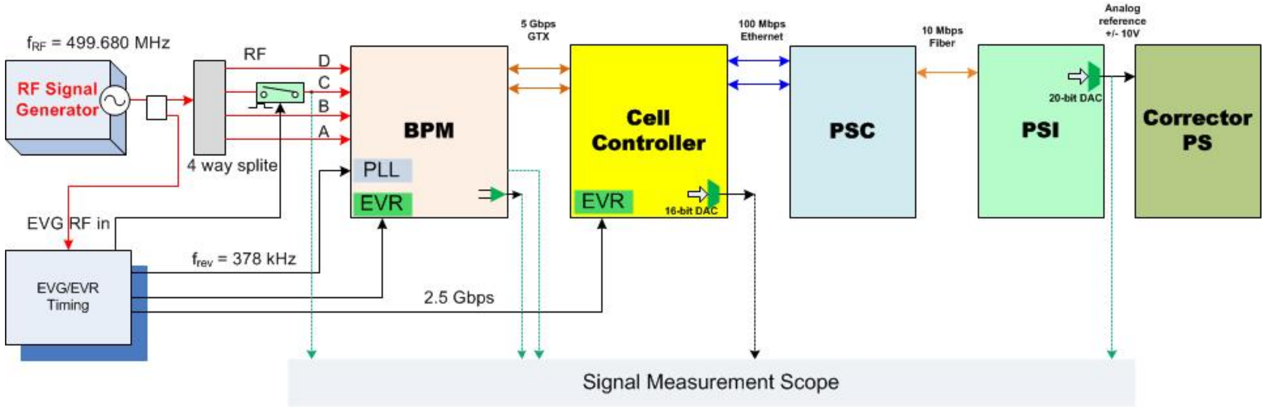

Ralph J. Steinhagen, Feedback Control for Particle Accelerators, PCaPAC 2016

# Fast Orbit Feedback System of NSLS-II




- Feedback rate : 10 kHz
- Bandwidth : ~ 200 Hz
- Control algorithm : SVD + PID
- Number of BPMs : 180 ea + ID bpms (27)
- Number of a fast correctors : 90 ea
- Communication update rate is 10 kHz
- All CC/BPM/AI/PS synchronized with timing

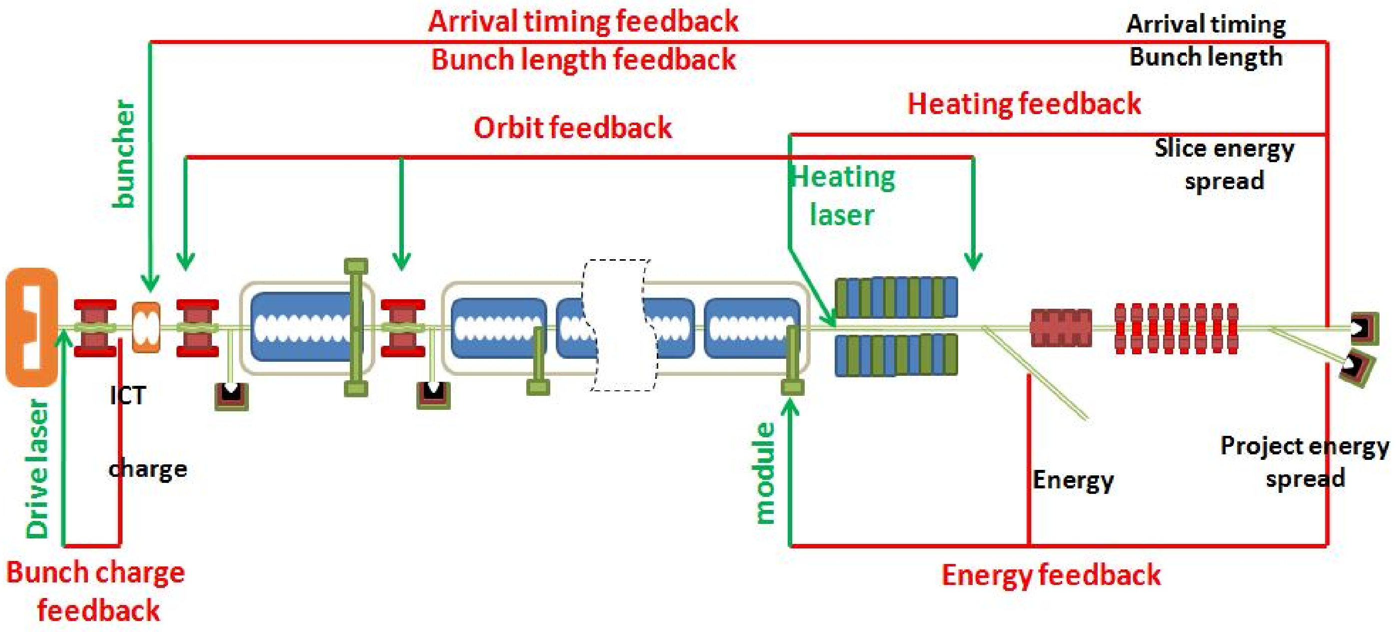



K. Ha, Y. Tian, MOC3O05, ICALEPCS 2015

# Feedbacks System of SHINE

- Transverse Loops: beam orbit, laser pointing (drive, heater and seed laser)
- Longitudinal Loops: bunch length, energy, charge, laser-electron relative time
- **Slow Feedback loops:**
  - Run at 1-10 Hz
  - Controlled by soft IOCs
  - Slow change parameters such as bunch length or laser position
- **Fast Feedback loops:**
  - Run at > 100Hz or kHz
  - Controlled by dedicated FPGA systems and low latency network
  - Fast actuators (air-cored coil) and RF amp/phase tuning
  - Fast orbit feedback ~ 1kHz

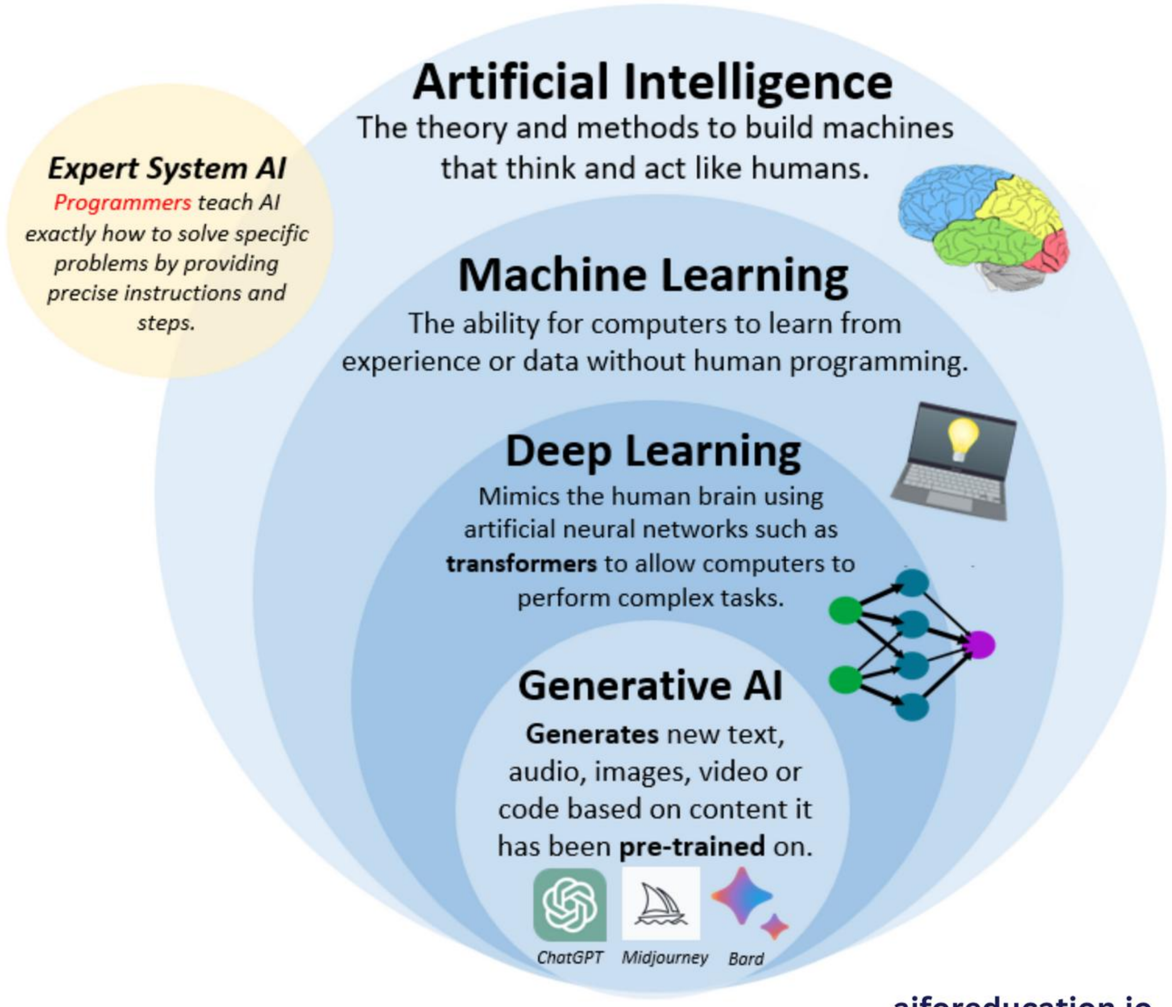



SHINE Injector

# Control with AI

# Artificial Intelligence

- **Artificial Intelligence (AI)** is the theory and development of computer systems capable of performing tasks that historically required human intelligence, such as recognizing speech, making decisions, and identifying patterns.
- AI is an umbrella term that encompasses a wide variety of technologies, including machine learning, deep learning, natural language processing ...

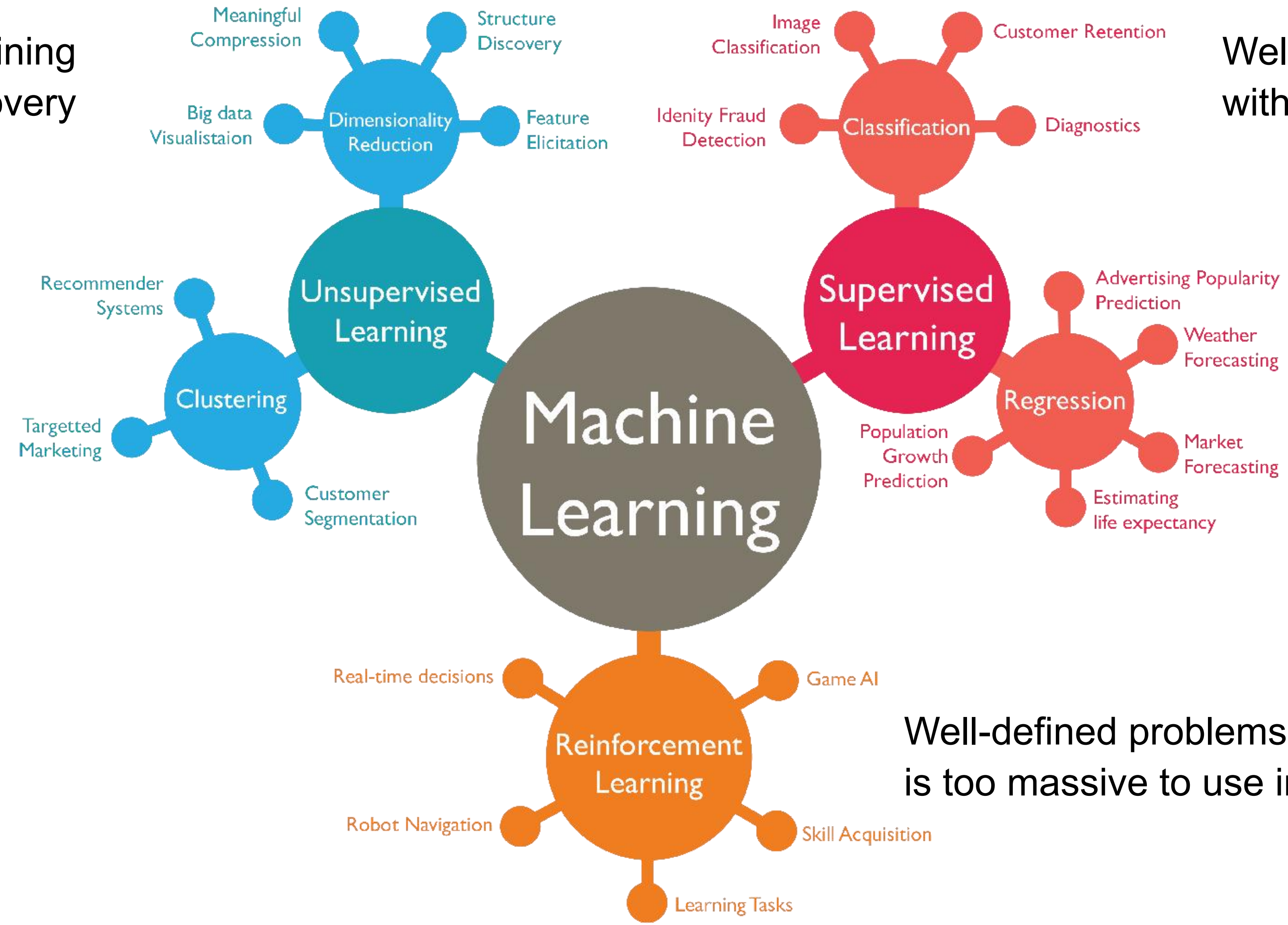



aiforeducation.io

# Artificial Intelligence

Data mining
pattern discovery

Well-defined problems
with small search space




Well-defined problems when search space is too massive to use in offline training

# Artificial Intelligence

## Tuning approaches can leverage different amounts of data/previous knowledge

less ← assumed knowledge of machine → more

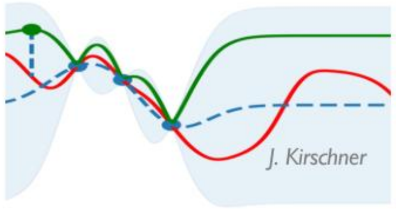
### Model-Free Optimization

*Observe performance change after a setting adjustment*

*→ estimate direction or apply heuristics toward improvement*

gradient descent
simplex

### Model-guided Optimization

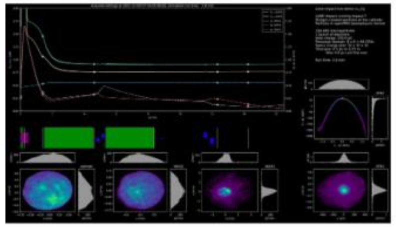



*Update a model at each step*

*→ use model to help select the next point*

Bayesian optimization
reinforcement learning

### Global Modeling + Feed-forward Corrections



*Make fast system model*

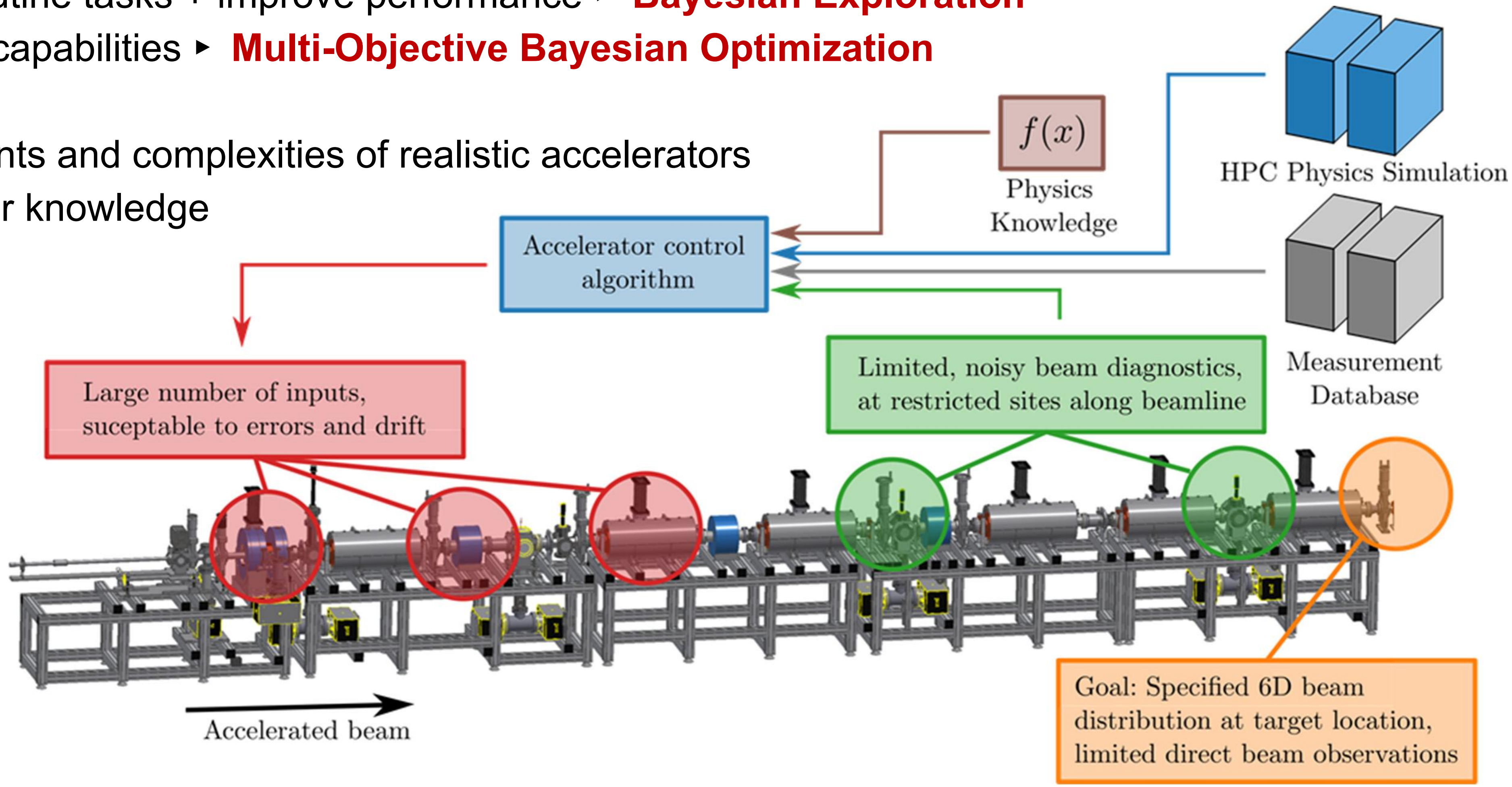
*→ provide initial guess (i.e. warm start) for settings or fast compensation*

ML system models +
inverse models

A. Edelen, Artificial Intelligence and Machine Learning for Particle Accelerators, NAPAC 2022

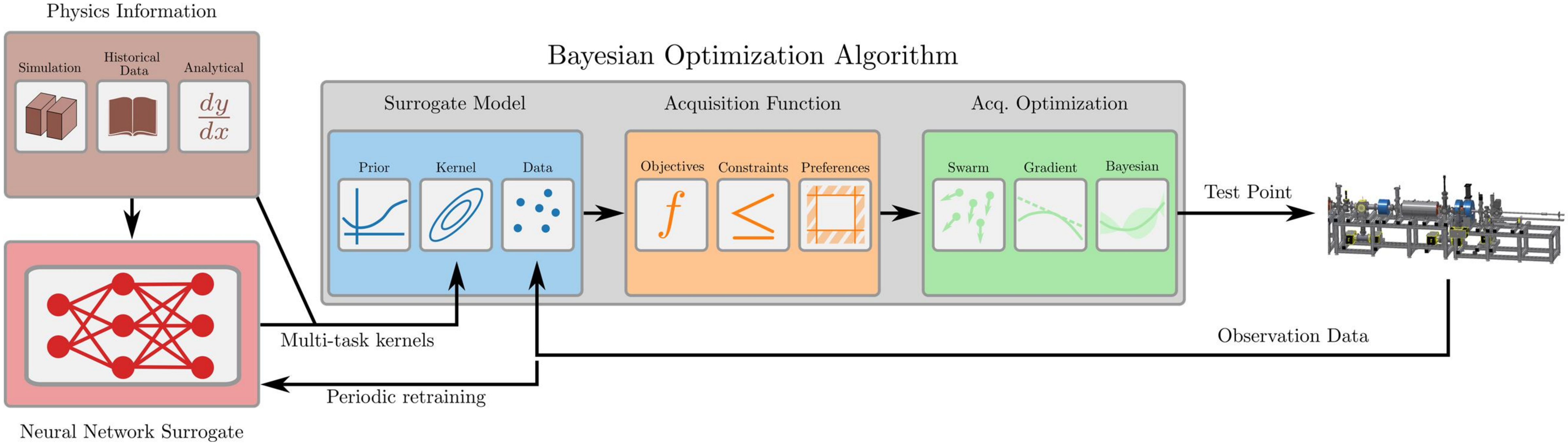
# Bayesian Optimization Based Accelerator Control

- Goals:
  - Automate routine tasks + improve performance ▸ **Bayesian Exploration**
  - Enable new capabilities ▸ **Multi-Objective Bayesian Optimization**
- Challenges:
- Practical constraints and complexities of realistic accelerators
- Incorporating prior knowledge
- Scaling




J. R. Roussel, A. Edelen, et al. Bayesian Techniques for Accelerator Characterization and Control, ICALEPCS 2021

# Bayesian Optimization Based Accelerator Control



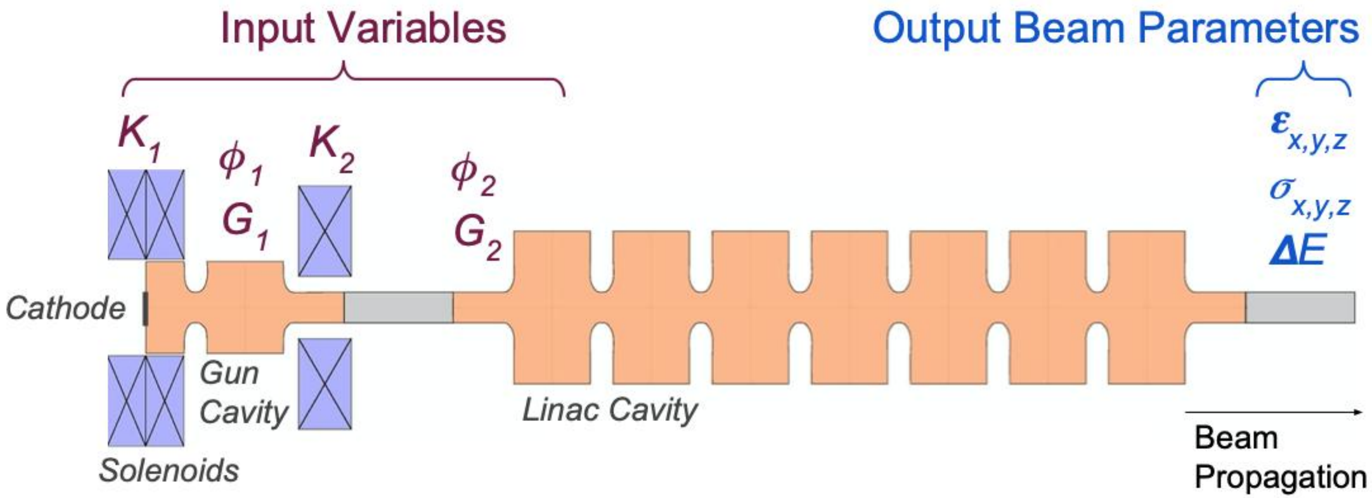

Bayesian exploration and hysteresis modeling represent steps towards a unified characterization and control system for accelerators

J. R. Roussel, A. Edelen, et al. Bayesian Techniques for Accelerator Characterization and Control, ICALEPCS 2021

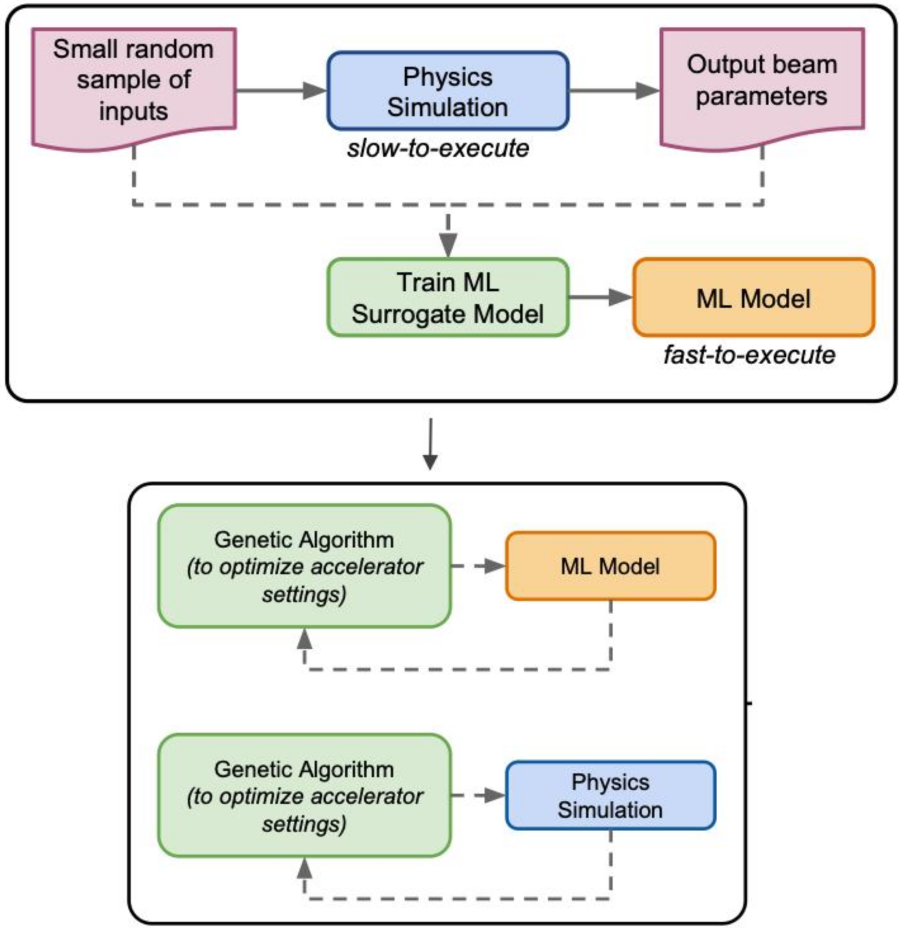
# Multi-Objective Photoinjector Optimization

*Test Case: Argonne Wakefield Accelerator Injector*

| OPAL simulation (PIC) : 3D space charge 3D field maps | NSGA-II for optimization: 200 generations ~350 individuals | 100 - 5K random points for training |
| --- | --- | --- |




Generate ML Model using Sparse Random Sample

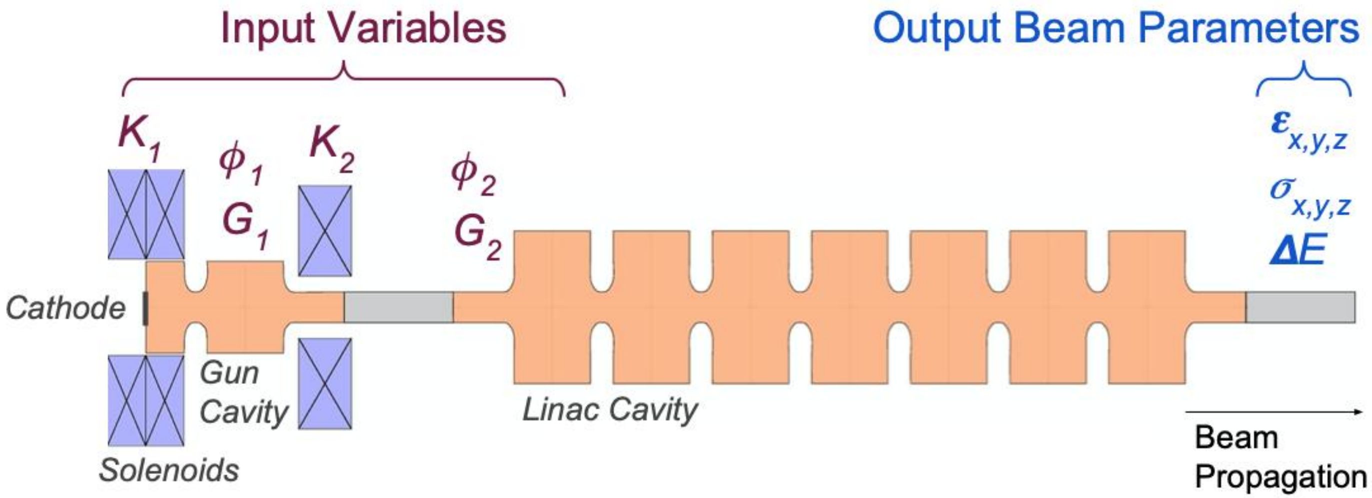



A. Edelen, et al. PRAB 23, 04601 (2020)

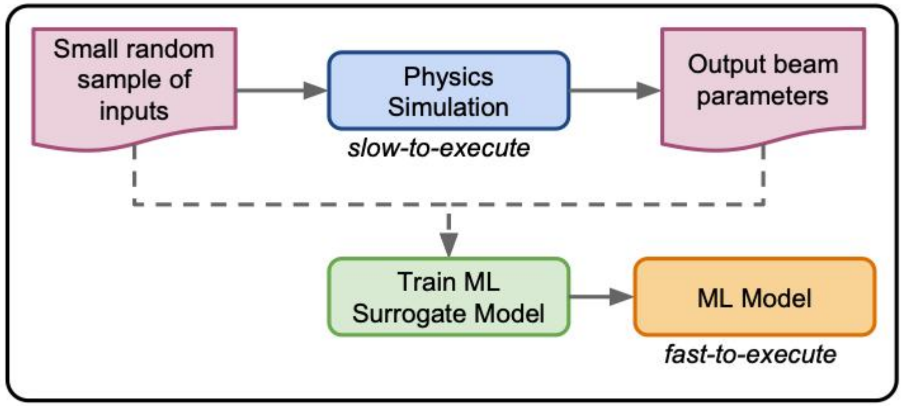
# Multi-Objective Photoinjector Optimization

*Test Case: Argonne Wakefield Accelerator Injector*

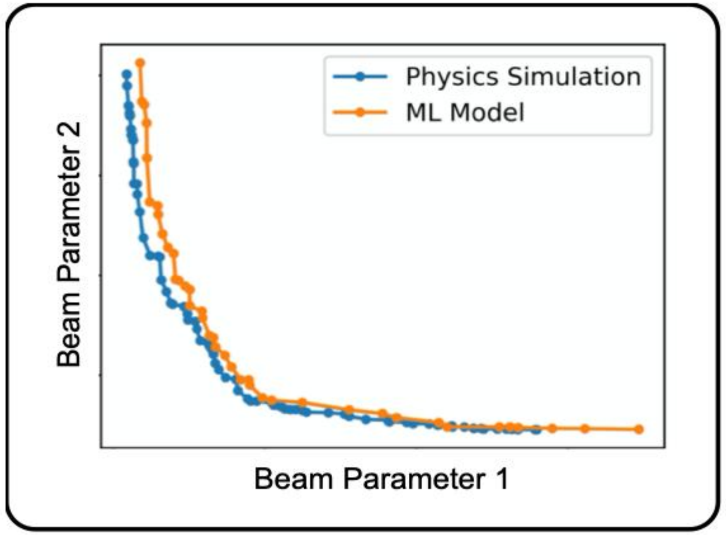
| OPAL simulation (PIC) : 3D space charge 3D field maps | NSGA-II for optimization: 200 generations ~350 individuals | 100 - 5K random points for training |
|---|---|---|




Generate ML Model using Sparse Random Sample




Compare Resulting Pareto Fronts

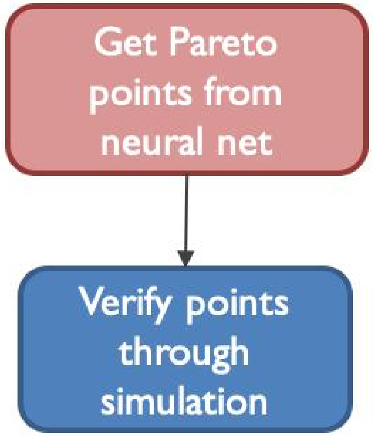


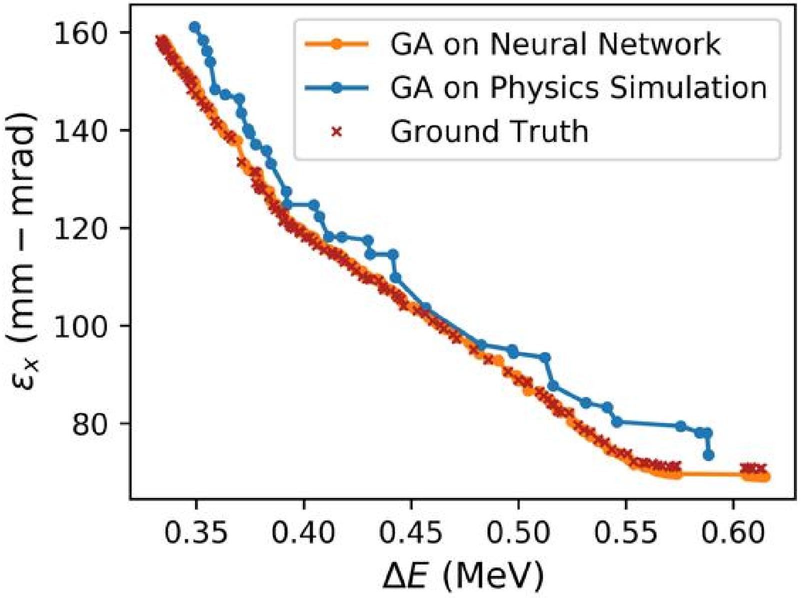

A. Edelen, et al. PRAB 23, 04601 (2020)

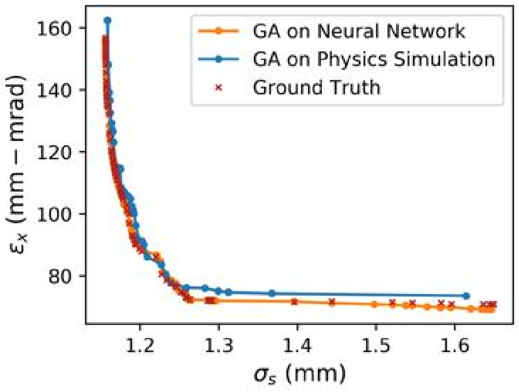
# Multi-Objective Photoinjector Optimization

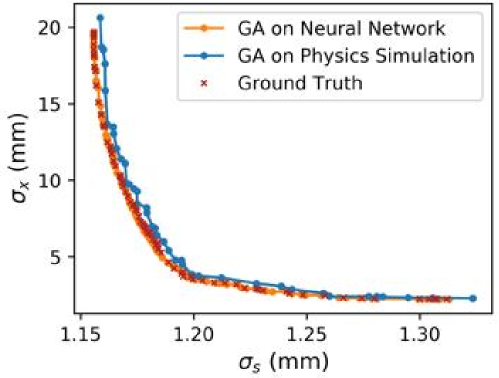
In some cases, optimization over simulation takes too long to converge

→ *validate Pareto front from neural network more directly*



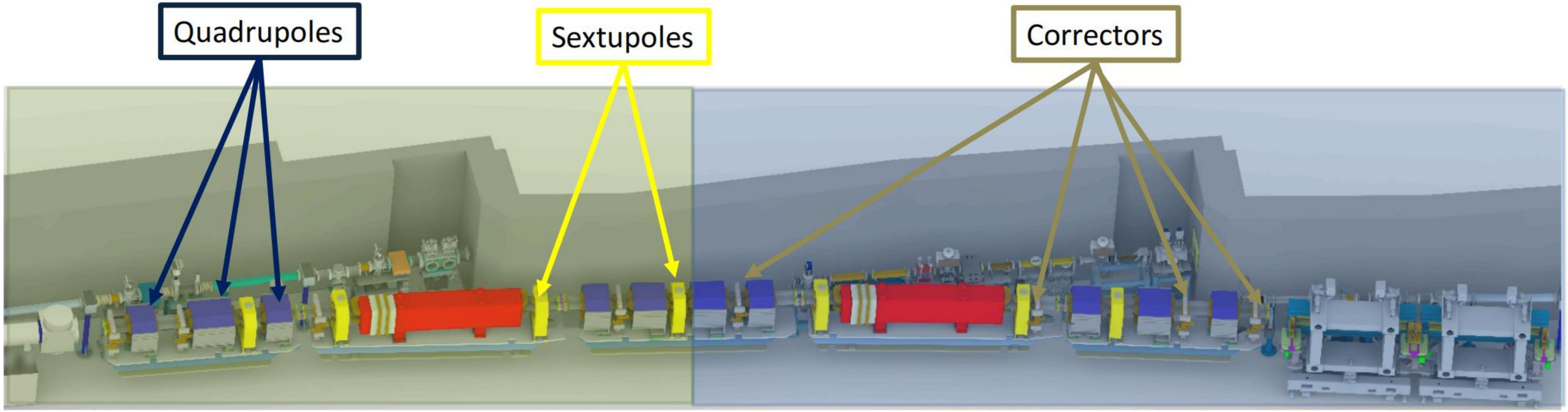




Required **130x fewer simulations** and had **$10^6$ times faster execution** in the optimization

A. Edelen, et al. PRAB 23, 04601 (2020)

# Anomaly Detection

- Detecting faulty magnet power supplies in the APS
- Can we predict if a fault will occur?
  - If yes, can we predict which magnet will fault
- Components of interest
  - 1320 magnet power supplies / 40 sectors (each has A (green) and B (blue) sections)

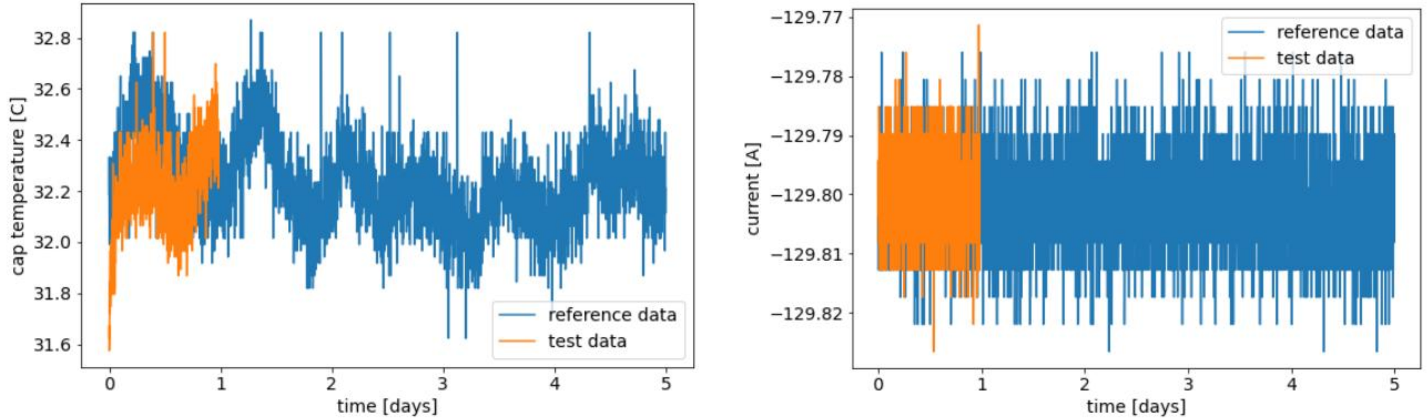


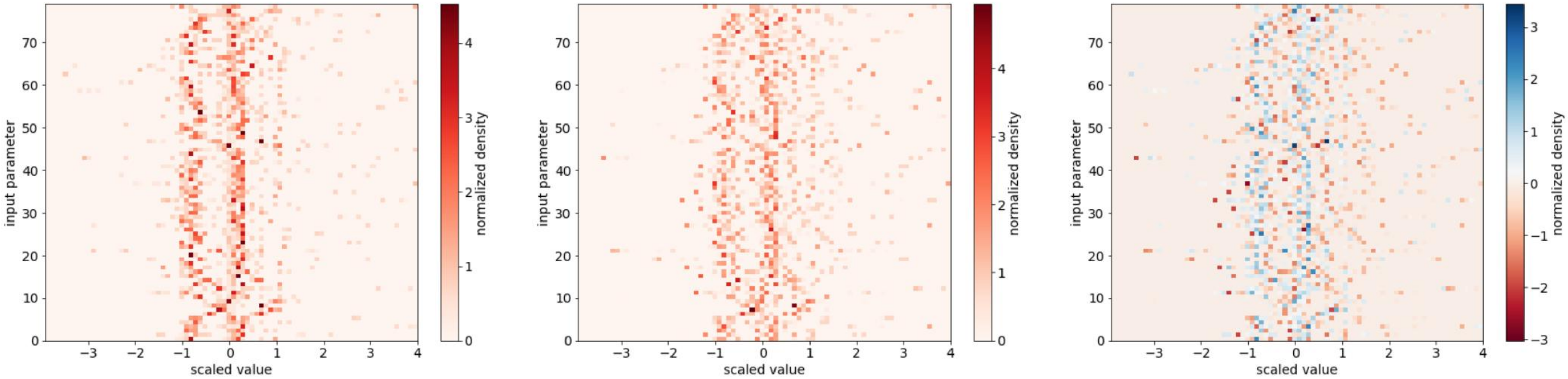

Jonathan Edelen, Kevin Bruhwiler, et al. Automating Particle Accelerator Operations with Machine Learning, ICALEPCS 2021

# Data collected over three years of operation

- Time series data for 1320 magnets
  - Power supply cap temperature
  - Current
  - Magnet temperature
- Data is aggregated by sector
  - Reference data (left) used for training and validation
  - Test data (middle) with known anomalies
  - Histogram difference (right)

Reference data (blue): no fault occurs in vicinity, normal operations. Test data (orange): magnet failure occurs; data is clipped and does not include final minutes prior to magnet fault.



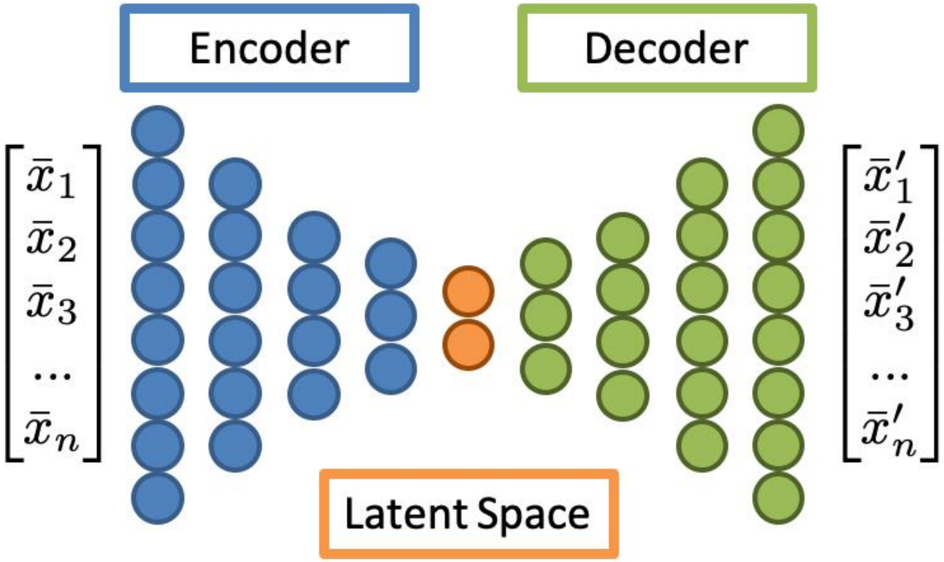





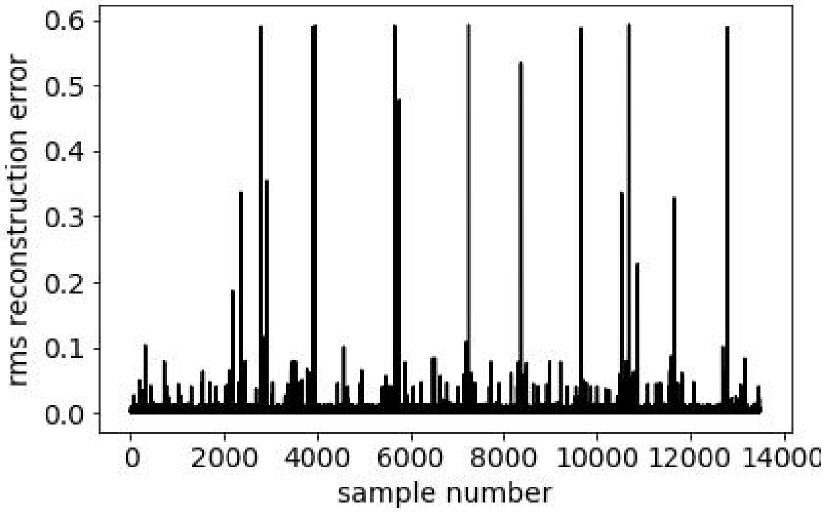

Jonathan Edelen, Kevin Bruhwiler, et al. Automating Particle Accelerator Operations with Machine Learning, ICALEPCS 2021

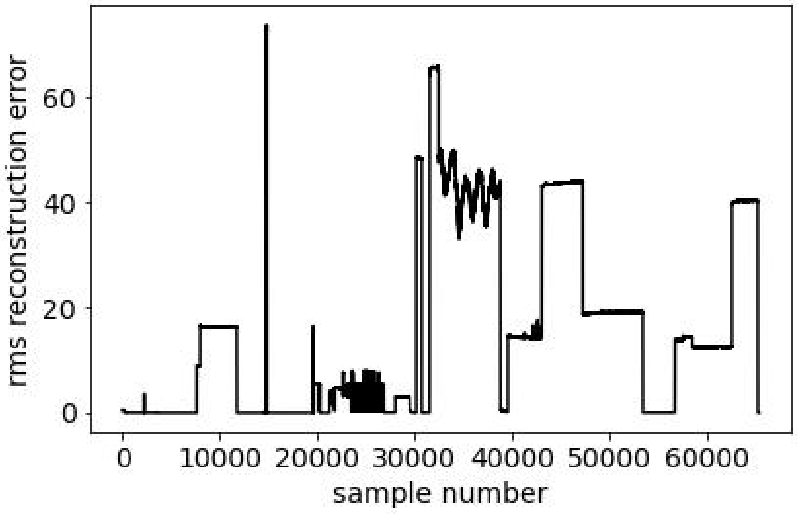
# Anomaly Detection

- Reconstruct unknown data using an autoencoder
  - Train and validate the autoencoder on known good datasets
  - Test on unknown data (may be good or bad)
  - Measure the degree to which the autoencoder successfully reconstructs the unknown data




$$\Sigma_n(\bar{x}_i - \bar{x}'_i)^2$$




Jonathan Edelen, Kevin Bruhwiler, et al. Automating Particle Accelerator Operations with Machine Learning, ICALEPCS 2021

# Summary

# If you want to know more ...

- International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS)
- This series of conferences facilitates fruitful collaborations among the world's control system specialists from particle accelerators, light sources, laser facilities, telescopes, tokamaks, etc.
- It started in 1987 and subsequently rotated between America (including North, Central and South America), Asia (including Oceania) and Europe (including Russia, the Near East and Africa).

**...**

**ICALEPCS 2021, hosted by SARI, Shanghai, China**

**ICALEPCS 2023, hosted by SARAO, Cape Town, South Africa**

**ICALEPCS 2025, hosted by APS/ANL, Chicago, USA**

**ICALEPCS 2027, hosted by KEK, Mito, Japan**

**...**

### Saturday September 20th Workshops

- EPICS Collaboration
- Tango
- PBCS: PLC Based Control Systems
- Towards Efficiency and Long-term Sustainability
- Controls GUI Strategies

### Sunday September 21st Workshops

- Control System Cybersecurity
- Bluesky
- Motion Control and Robotics
- Advancing AI/ML and Generative Models for the Control of Large Complex Systems
- Advanced Control (half-day)
- Safety Lessons Learned (half-day)

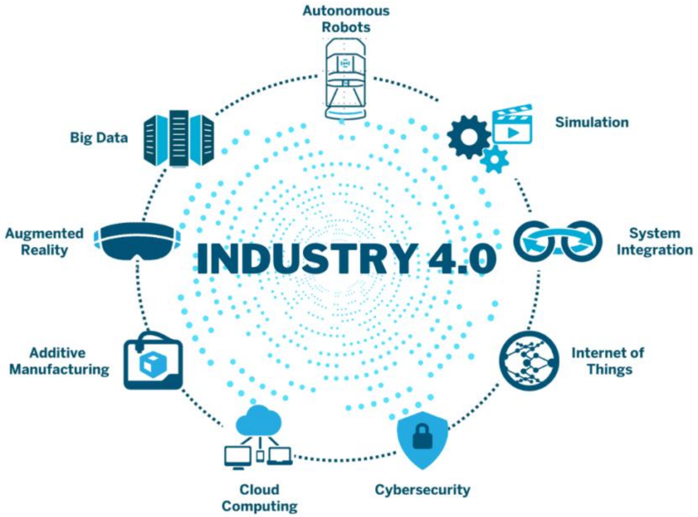
**Sep 20 - 26, 2025**

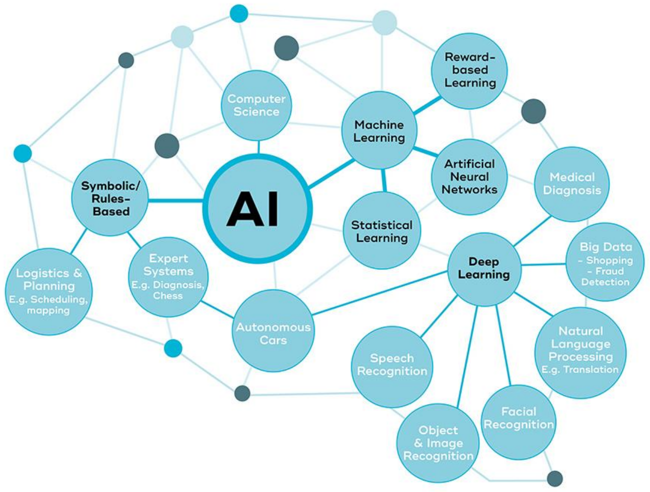
https://www.icalepcs.org/

# Summary

**The large-scale control system integration**

**requires extensive knowledge and comprehensive consideration.**

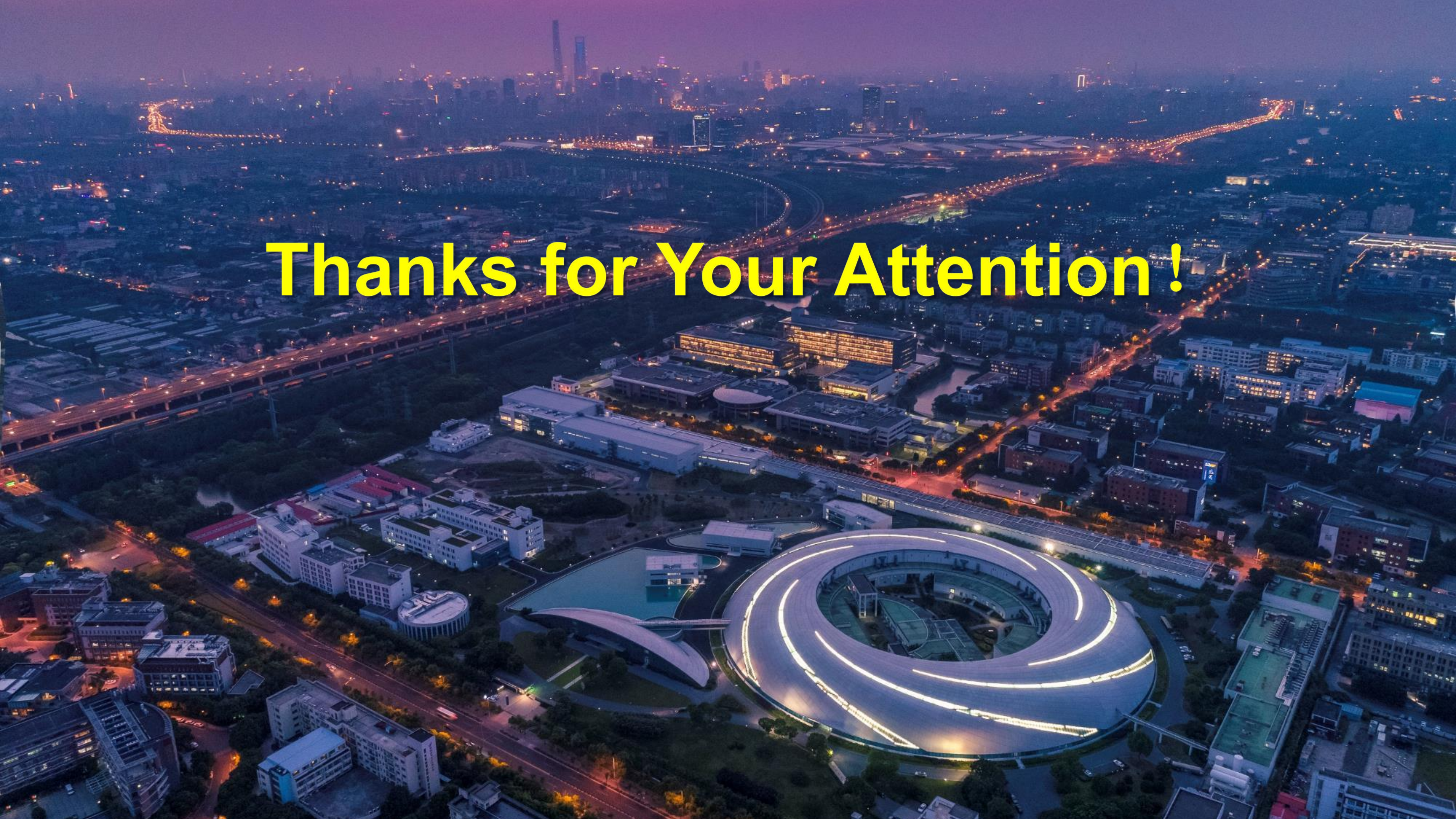




**more devices & more data** | **more intelligent** | **high availability**

https://www.calsoft.com/what-is-industry-4-0/ | https://qbi.uq.edu.au/brain/intelligent-machines/ | https://www.apixel.com.sg/


Thanks for Your Attention !