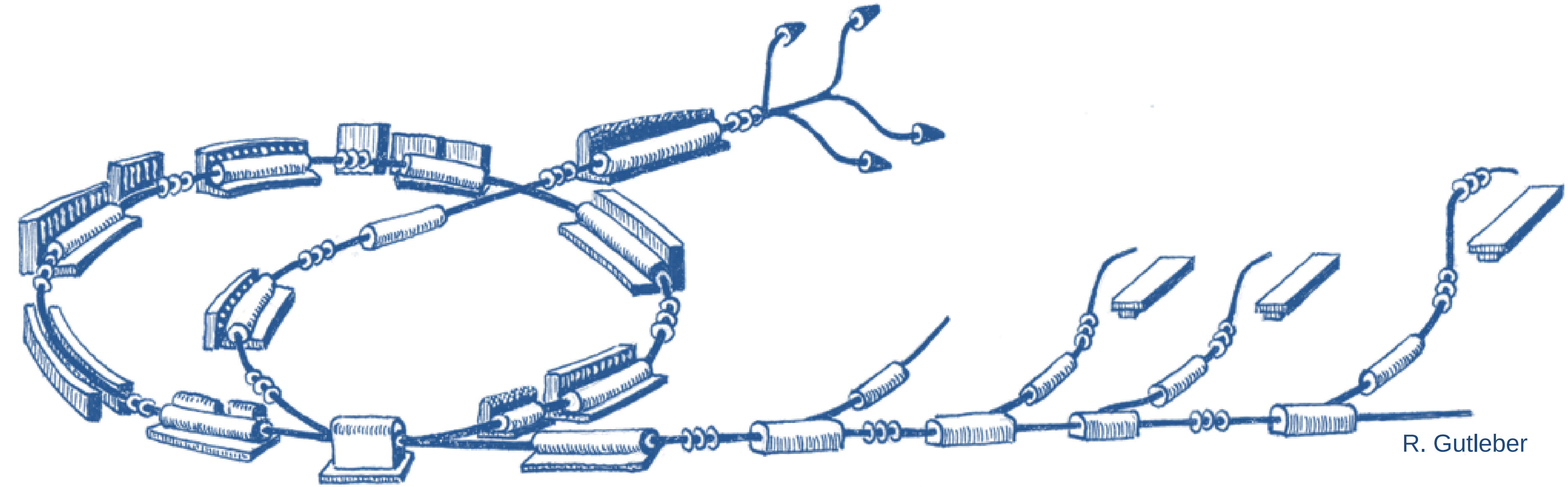
MACS Architecture

MACS Week

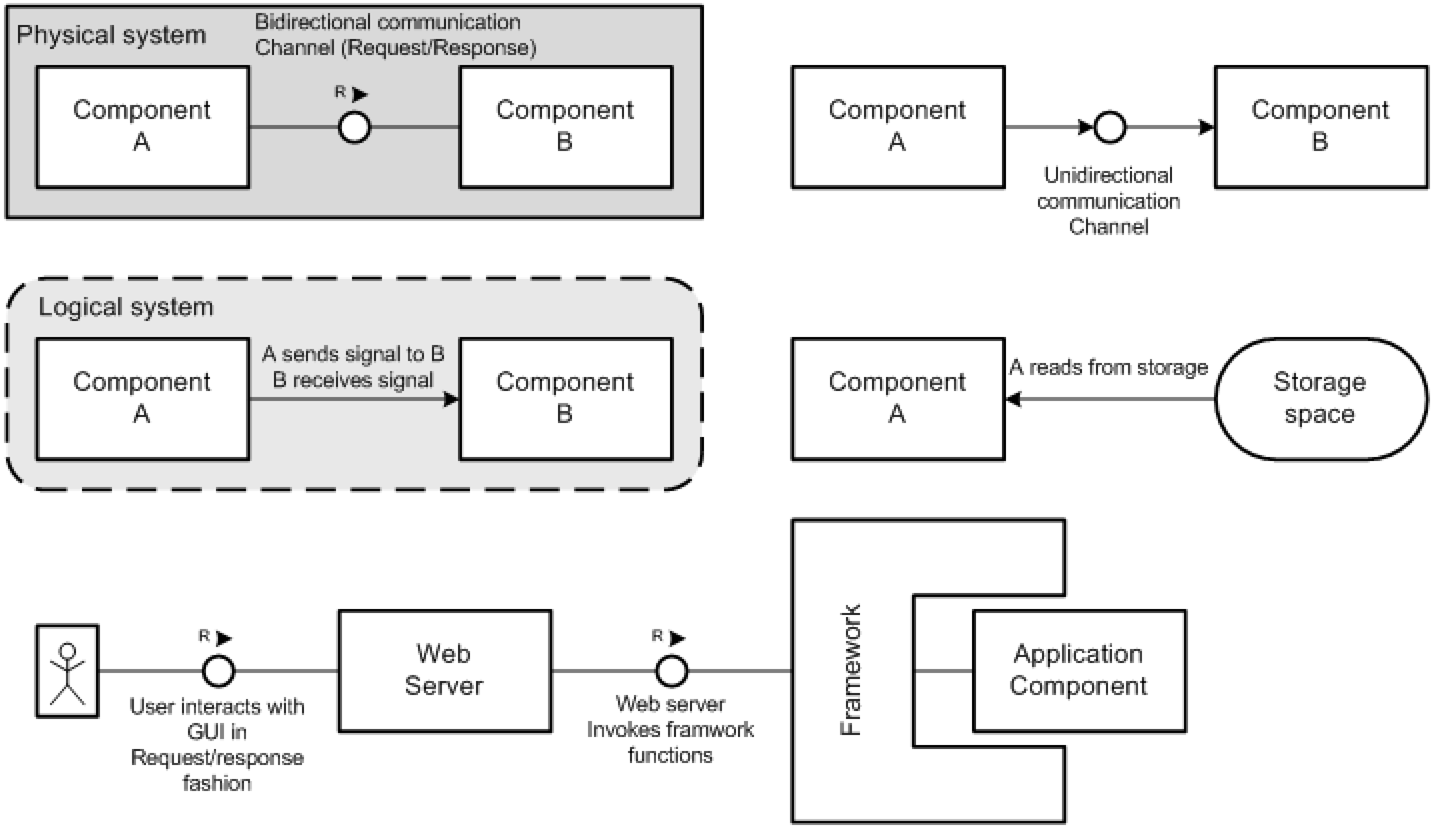
June 24th, 2010

Johannes Gutleber

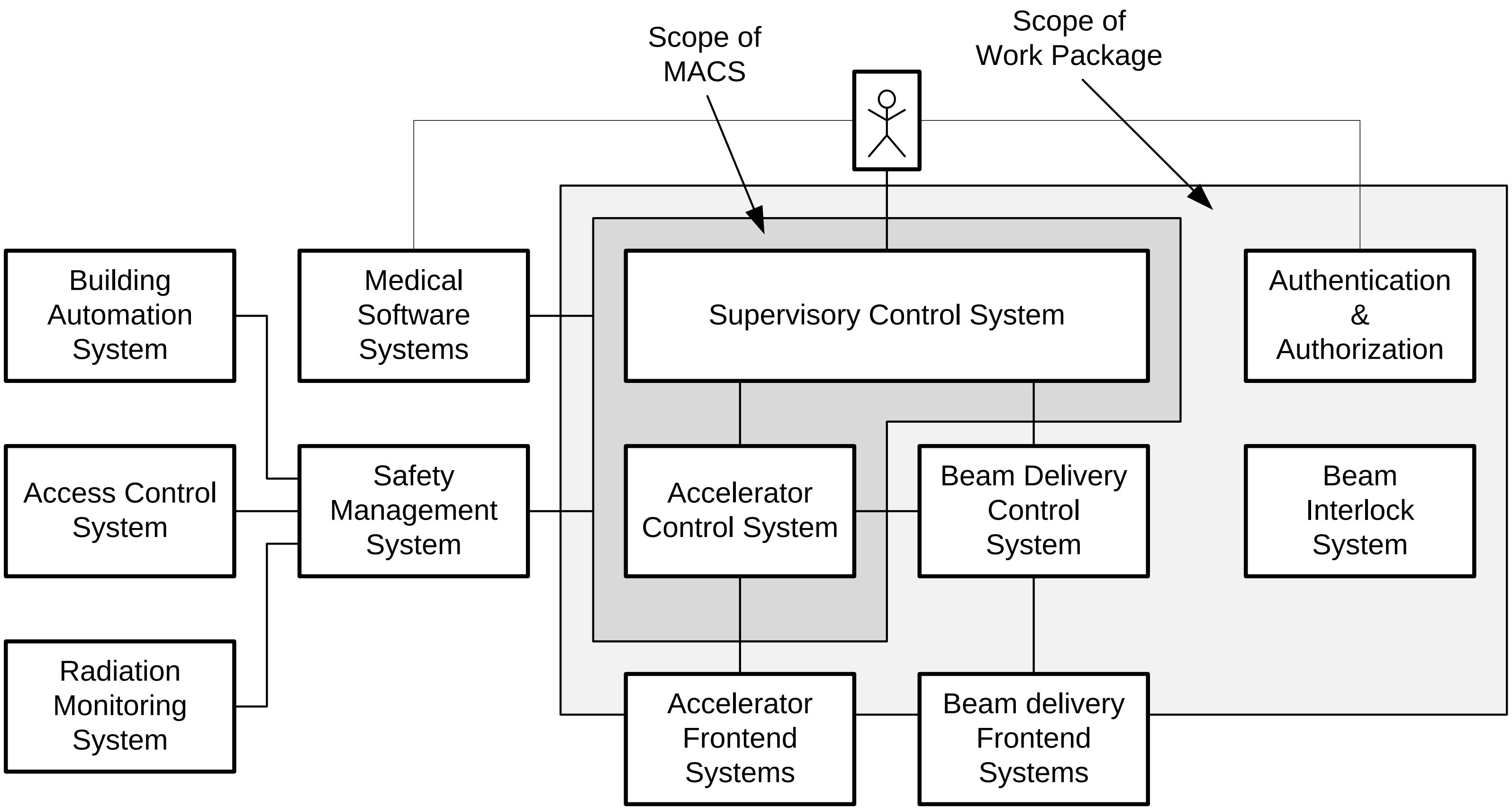


Functional Modeling Concept (FMC)

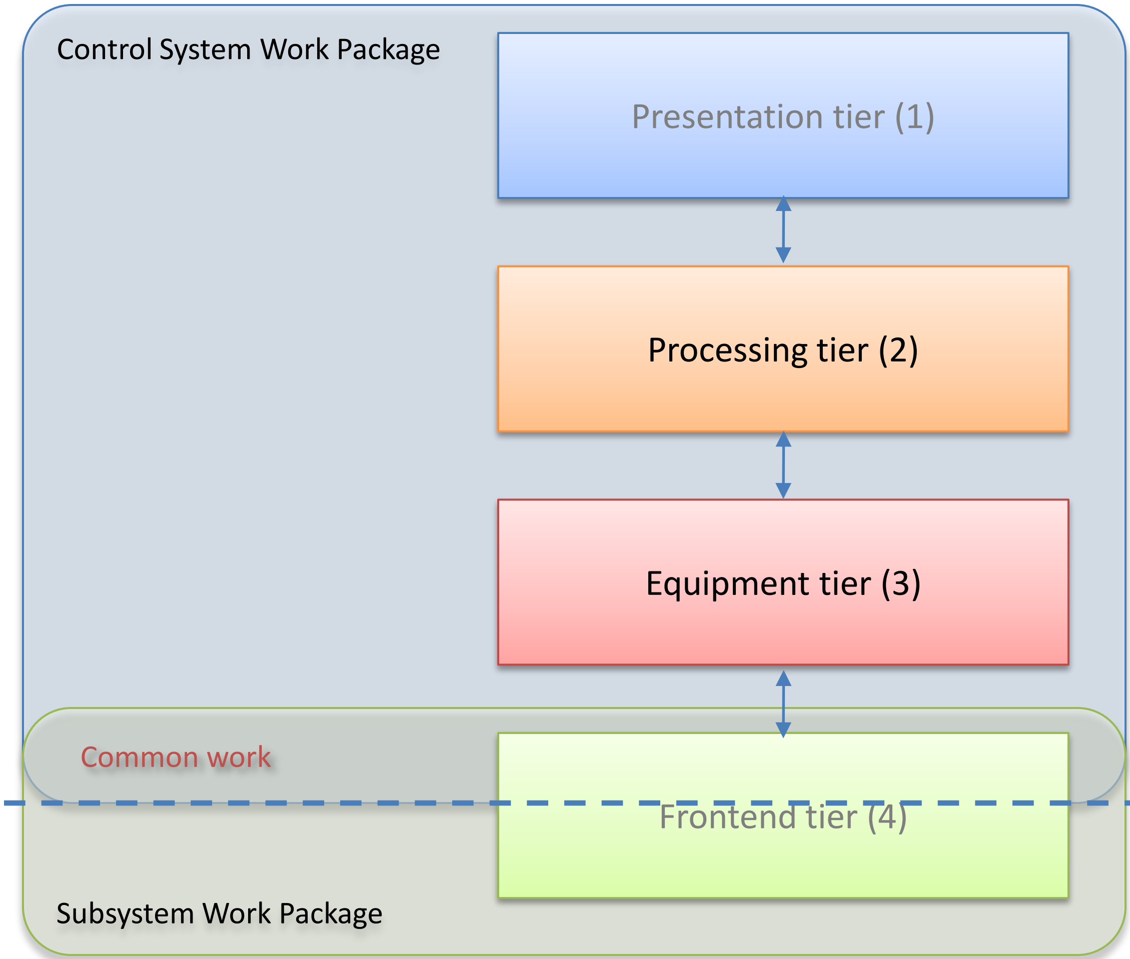
- Notation used to describe architecture



Scope


Scope of MACS
Scope of Work Package
Building Automation System
Medical Software Systems
Supervisory Control System
Authentication & Authorization
Access Control System
Safety Management System
Accelerator Control System
Beam Delivery Control System
Beam Interlock System
Radiation Monitoring System
Accelerator Frontend Systems
Beam delivery Frontend Systems

3 + 1 Tier Architecture



Graphical user interfaces
Remote operation

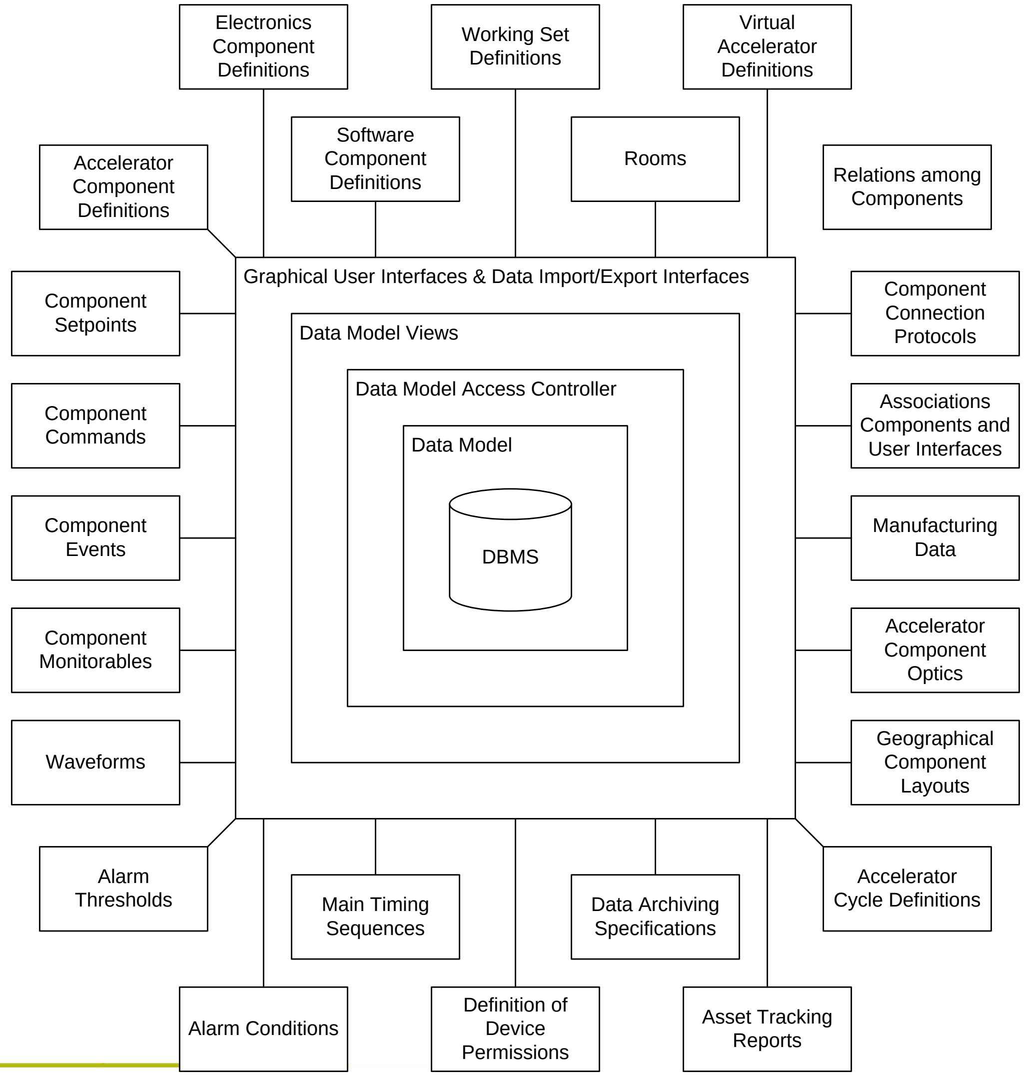
Configuration for devices
Supervisory state machines
Data logging, SCADA

Protocols and data format adapters, fan-out and data concentration, timing, SADS

Front-end adaptation and lean framework layer

Control logic and closed
Real-time control loops

Repository Management System

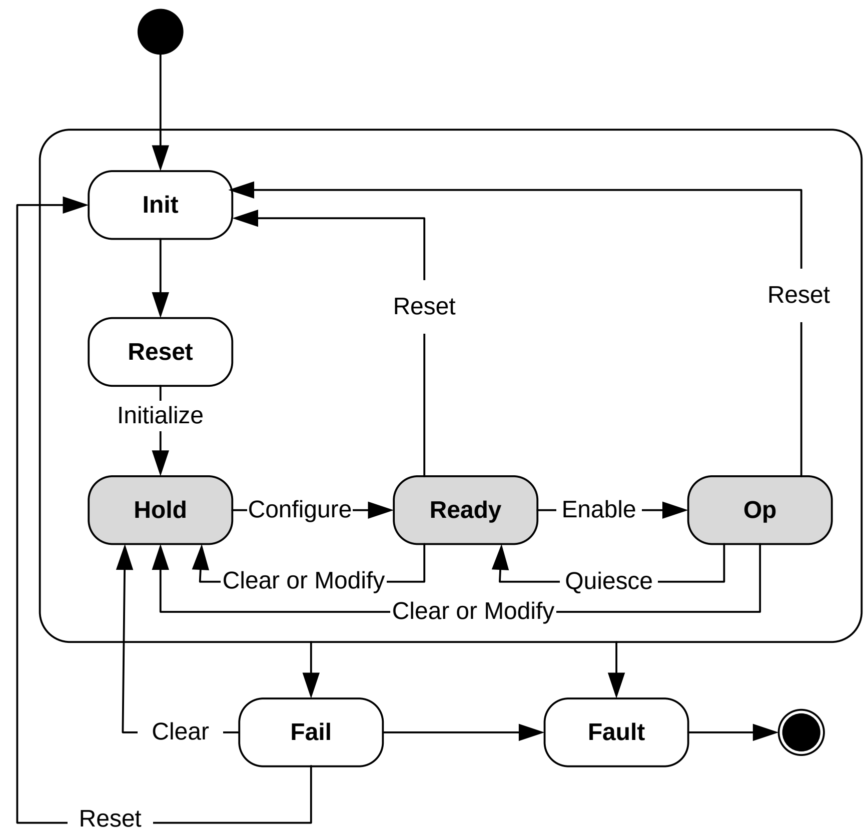


Component Model

- **Physical** accelerator **element** or assemblies of physical elements are **represented in software, called component**
- Each component has
  - Unique identifier
  - Name
  - Properties
  - Ownership at runtime
  - State machine
  - May react to timing events (cycle dependent operation)

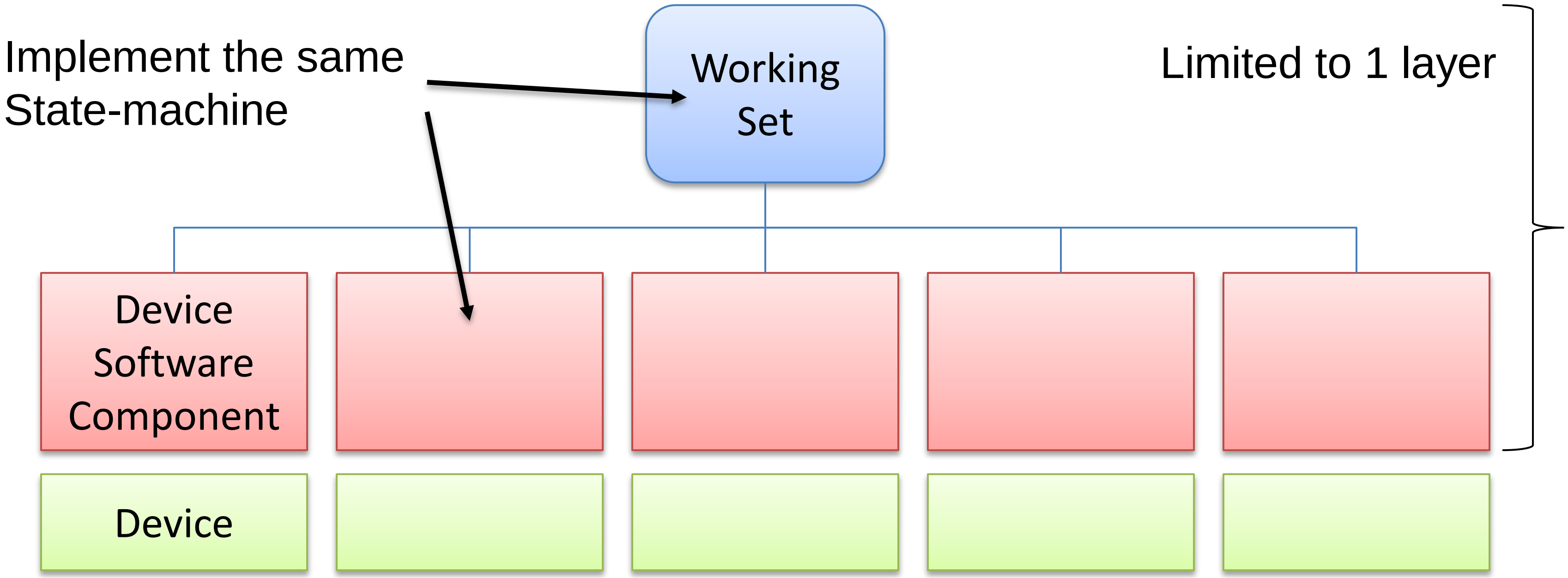
Uniform State Machine

- Every device in 1 of **4 modes**
  - **Service, Physics, QA, Clinical**
- Every device **implements all states and transitions**
- **Additional**, device specific **commands possible** in “Op” state
- State machine implemented in “software representation” of device
  - Front End Controller (**FEC**)



Working Sets

- Working Set is a **collection of** defined **components**
- Can be controlled as a "single virtual device"
- Can be assigned a "single set of configuration parameters"

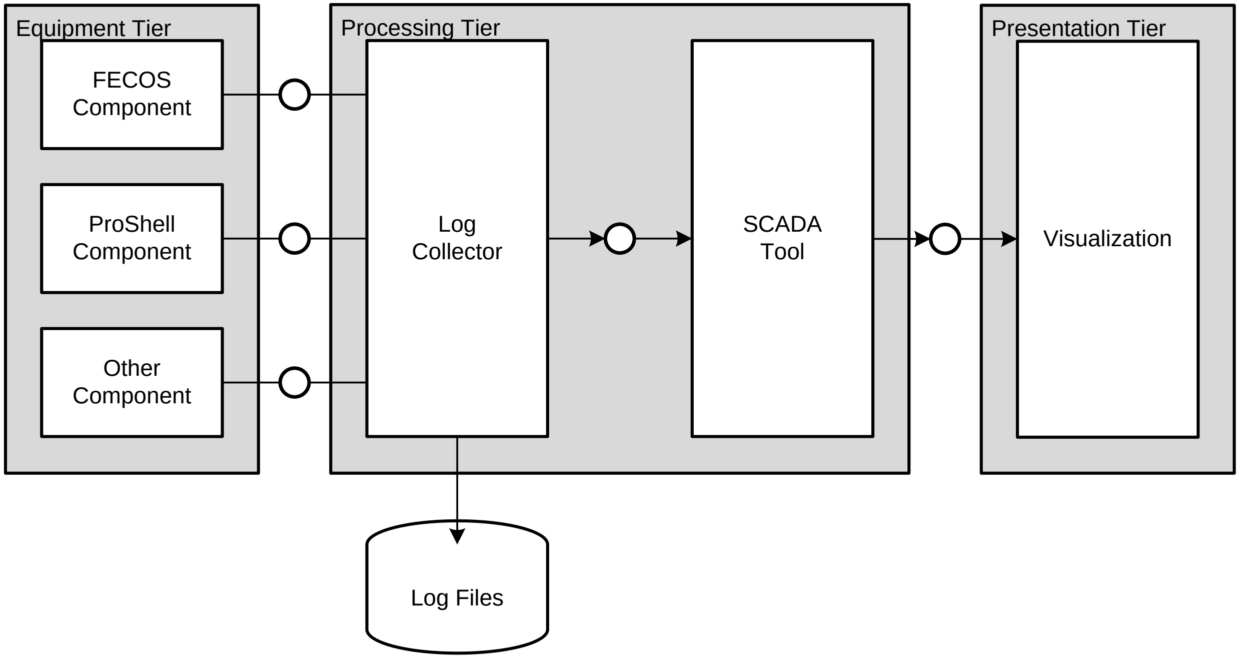


Errors and Alarms

- **Devices** (hardware and software) limit themselves to **provide information** that can be processed by SCS to generate alarms
- **Alarms** are **defined in PVSS** based on numeric quantities and Boolean expressions
- "**Failed**" and "**Fault**" **states** for each device
  - Device must stop working (fail-silent behaviour)
  - Possibility to recover from "Failed" via operator intervention
  - Alarm raised by SCS in either case that must be acknowledged
- Error during processing a command is notified to SCS
  - Not a failure! Alarm can be defined by operator at SCS level is needed
- QoS indicator for each "software" service

Logging (Execution)

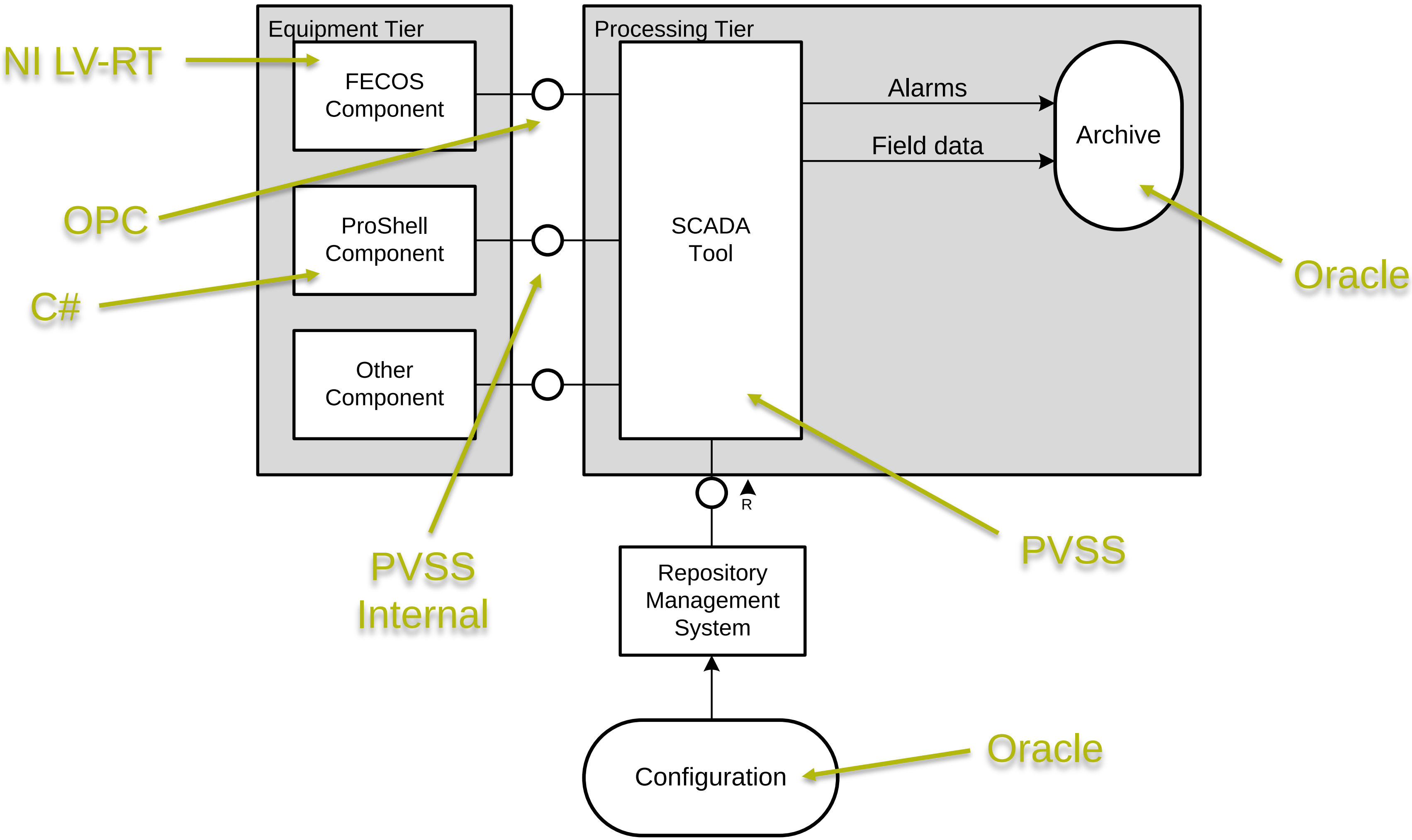
- Process of tracking changes in execution of a program
- Rely on standard protocol, data format and tools
  - Log4Net tool, Log4J protocol, all logs end up in PVSS



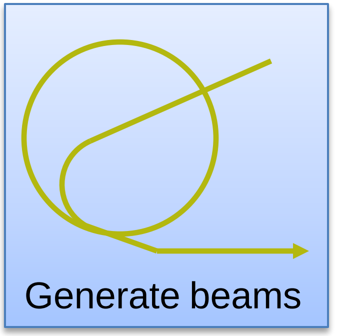
Data Logging

- **Automated process** of continuously collecting and archiving process measurements on a timely basis.
- O(3000) values collected at average frequency of O(1) Hz.
- Carried out by PVSS SCADA tool
- For each data item **conditions** can be specified that determine **when** the value is **collected or archived**.
- **Limited transformations** (e.g. unit conversion) may be applied to the gathered values in order to record a set of derived data values.
- **Archiving** can be configured for each data item independent of the time when it is collected
  - tradeoff between storage space efficiency and measurement accuracy

Data Logging Architecture

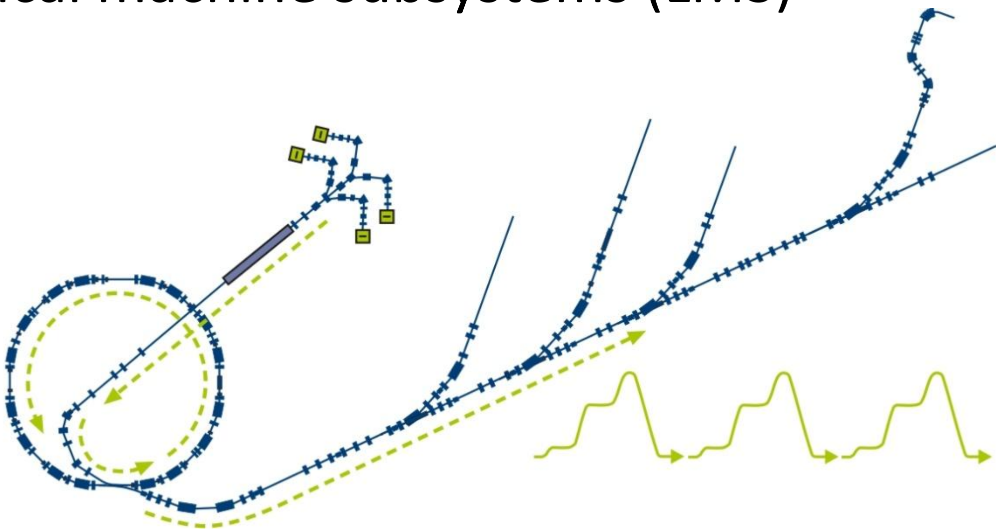


Beam Generation Process



Operation Principle

- Accelerator is split into logical machine subsystems (LMS)
  - Sources
  - LEBT
  - Linac
  - MEBT
  - Main Ring
  - Extraction Line
  - Individual irradiation lines



- **Virtual Accelerator (Vacc)** comprises at least 1 LMS working set
- VAcc works on a "**cycle**" basis

Cycle Definition

- Accelerator cycle **describes beam characteristics to be produced** during a period of time by specifying a **set of configuration parameters**.
  - What (parameters)
  - When (timing)
- Generation of cycle causes activation of pre-defined actions at pre-defined, cycle-dependent times in front-end software components that control the physical elements that constitute the accelerator.
- At each point in time, a single cycle is active per VAcc
- A defined cycle can be played over a “VAcc”.

Cycle Definition Contents

- Ion type
- Source type (optional, allows binding of cycle to source)
- Target room of beam (optional, for clinical mode)
- Energy
- Beam dimension
- Intensity
- Spill length

Cycle Code

64 bit wide identifier to which configuration parameters can be associated for each front end controller

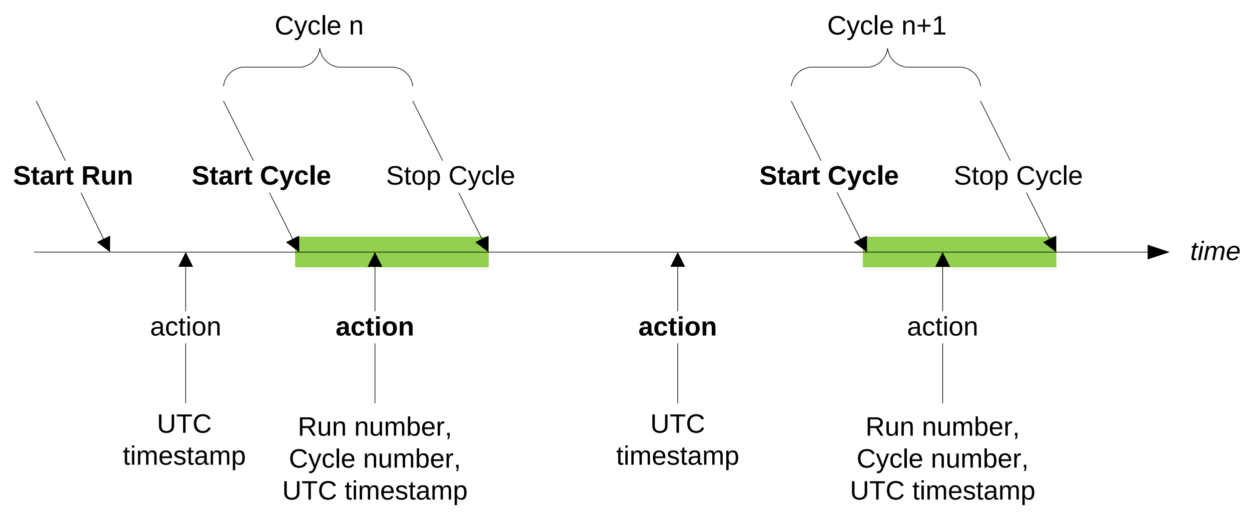
Bit	Size	Name	Description
0-15	16	Energy	*n* x 0,1 MeV from 0 to 800 MeV (**8000 steps**)
16-18	3	Particle species	0 unused, p, C and space for other light ions. **7** in total
19-21	3	Source number	0 means unbound, **7 sources if** indication **needed**
22-24	3	Degrader	0: 0%, 1: 10%, 2: 20%, 3: 50% (**3 values**)
25-28	4	Spill length	0,1; 0,2; 0,5; 1, 2, 3, 4, 5, 6, 7, 10 sec (**11 values**)
29-31	3	Beam line	0 means unbound, **up to 7 if** indication **needed**
32-35	4	Beam x size	Size in mm from **1 to 10**
36-39	4	Beam y size	Size in mm from **1 to 10**
40-41	2	Mode	0: **service**, 1: **physics**, 2: **QA**, 3: **clinical**
42-43	2	Code format	0: current version, others for future format changes
44-47	4	Minor version	Intermediary versions for MD, modes 0 (S) and 1 (P).
48-63	16	Major version	Version indicates which settings to use for this cycle

MSB LSB

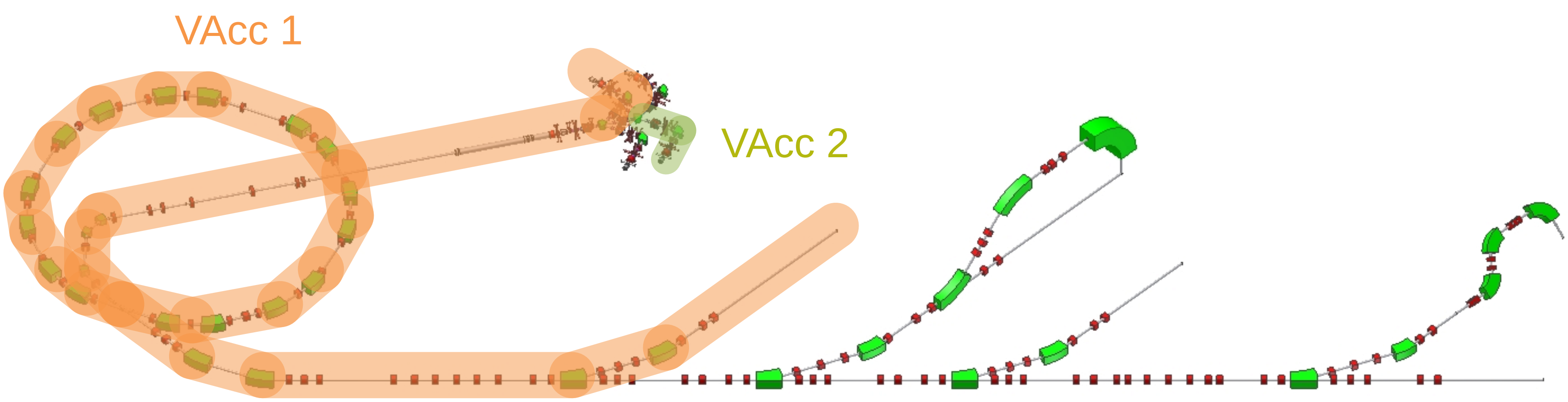
31	20	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Word
Beam line			Spill length				Degrader			Source			Particle			Energy																0
Major Version																Minor Ver.				Fmt		Mode		Y size				X size				4

Beam Stamp

- A record of information updated throughout the entire lifetime of the facility
- Unambiguously **associates in time any piece of information generated during operation to a cycle**
  - e.g. occurrence of a command, acquired data, alarms and errors
- Consists of a run number and a cycle number

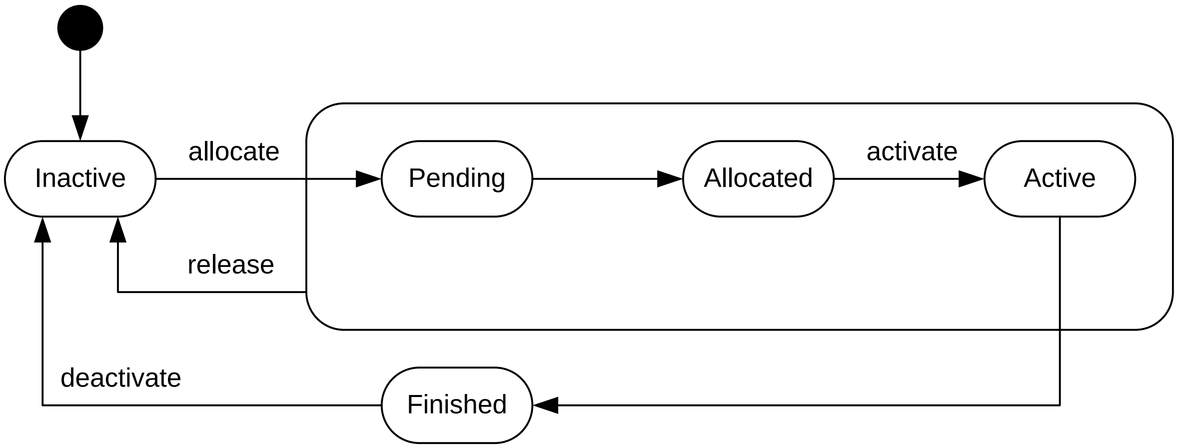


Virtual Accelerators



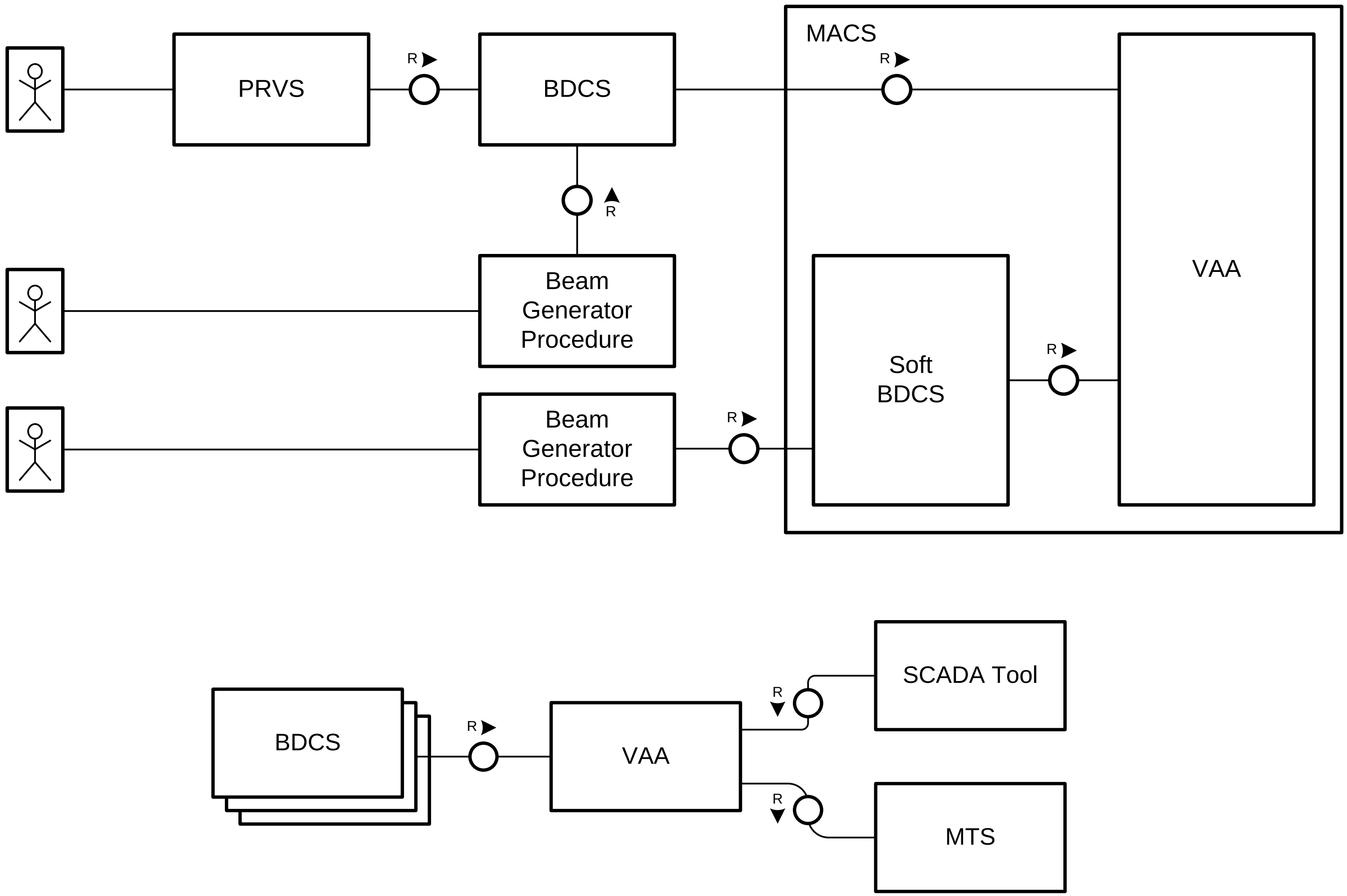
- Virtual accelerator (VAcc) is a collection of working sets (LMS)
- **Used for operation purposes** (beam generation)
- Freely definable using RMS
- Non-overlapping VAccs can be active concurrently
- If VAccs overlap, only 1 VAcc active at that time

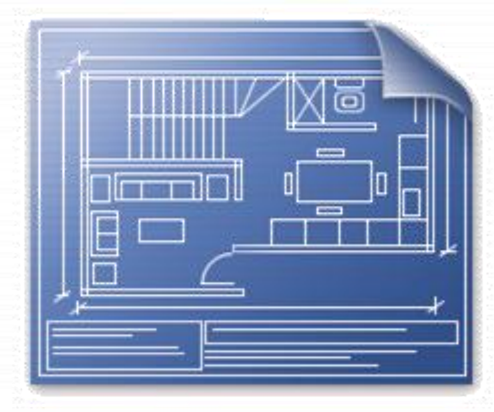
VAcc State Machine



- BDCS allocates a Vacc from Virtual Accelerator Allocator (VAA)
- If VAcc overlaps becomes “pending”
- If “allocated”, resources are owned by requestor
- If “active” VAcc is ready for beam generation
  - Beam interception devices and safety elements are “armed”
- If VAcc is released/deactivated, “pending” gets allocated

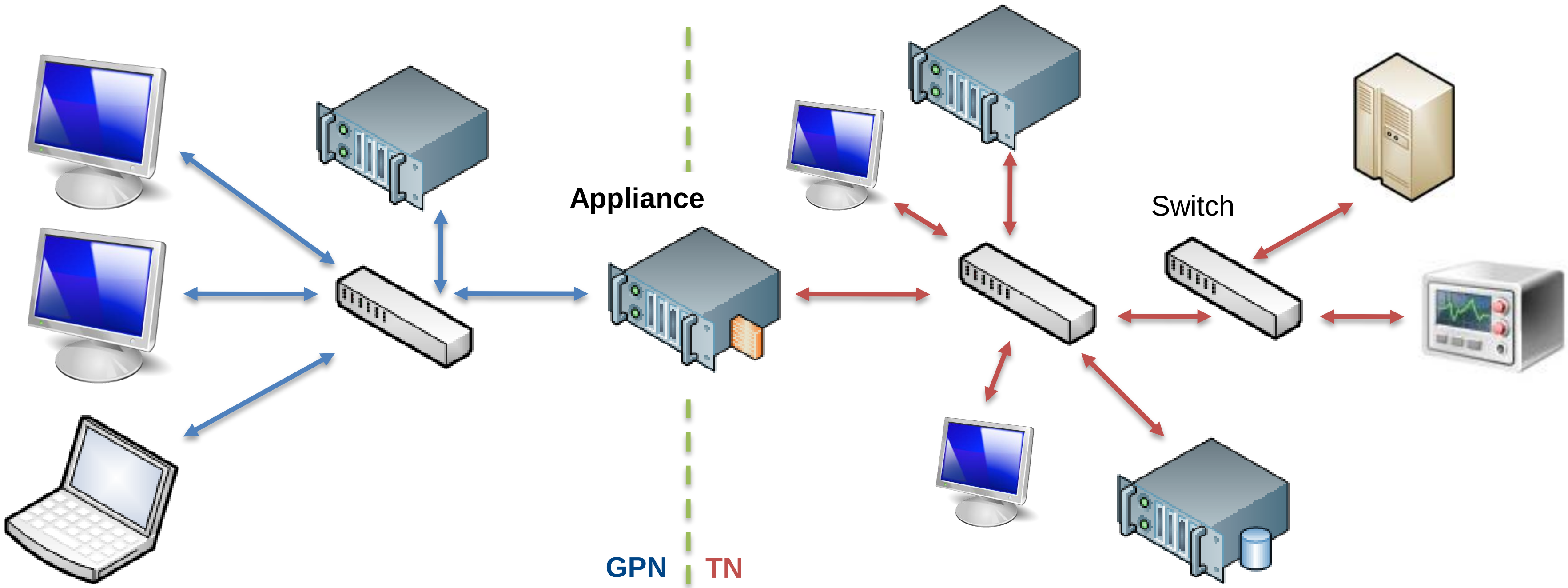
Beam Allocation





Blueprint of Supervisory and Accelerator Control System Components

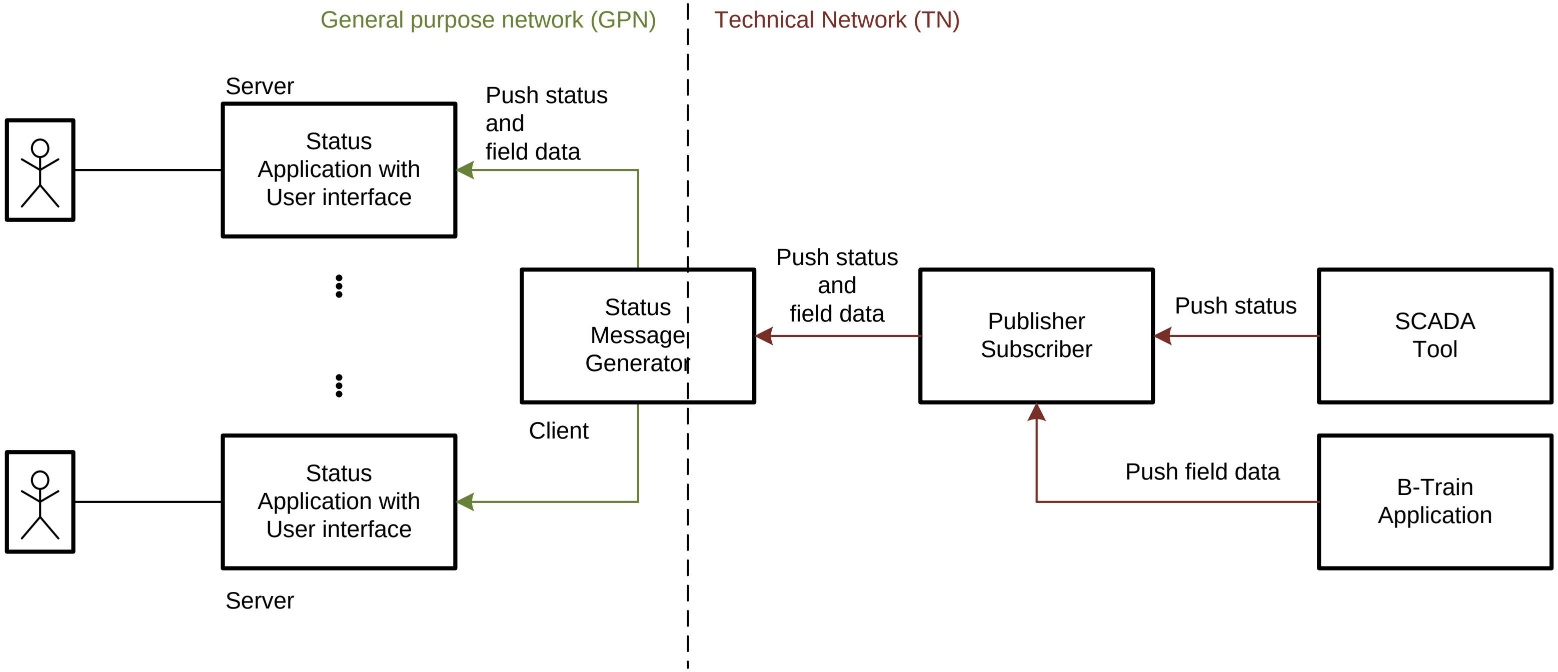
Network Organization



- SCS/ACS devices in physically separate technical network (TN)
  - Applies also to control room equipment
- General purpose network for office IT and private computers

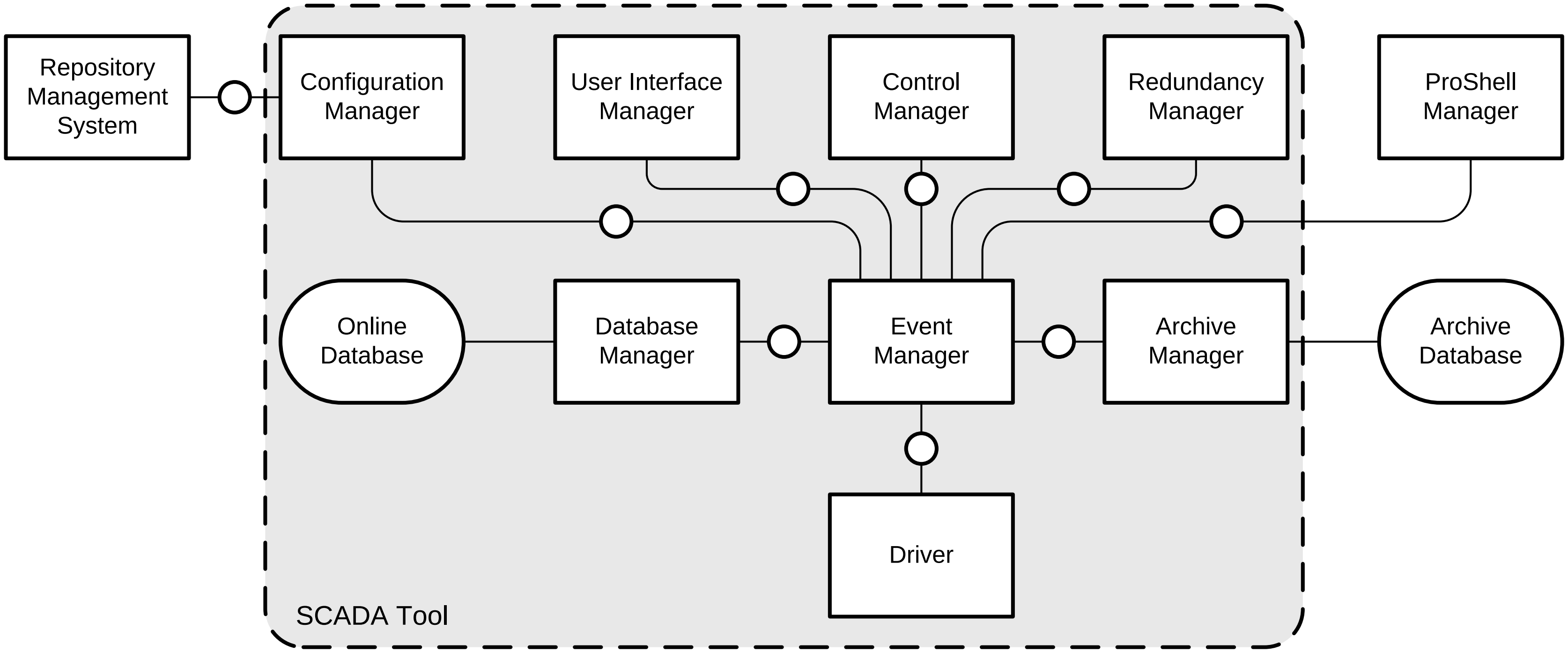
ebg MedAustron

Status Displays



- Current status of operation and near real-time update of cycle on various displays in the facility
- Available in techn. network and general purpose network

SCADA Tool



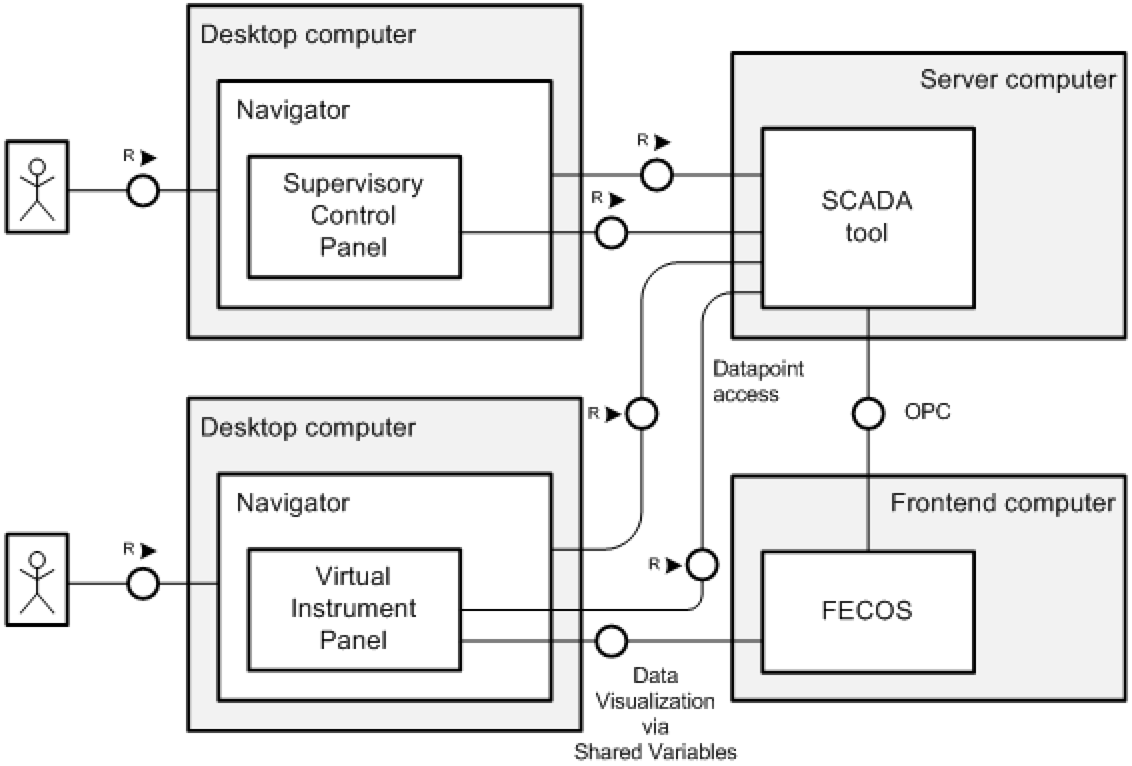
- PVSS to be configured from RMS database
- Keeps **state of entire system**
- **Single point of entry** to interact with all subsystems

Datapoint Concept

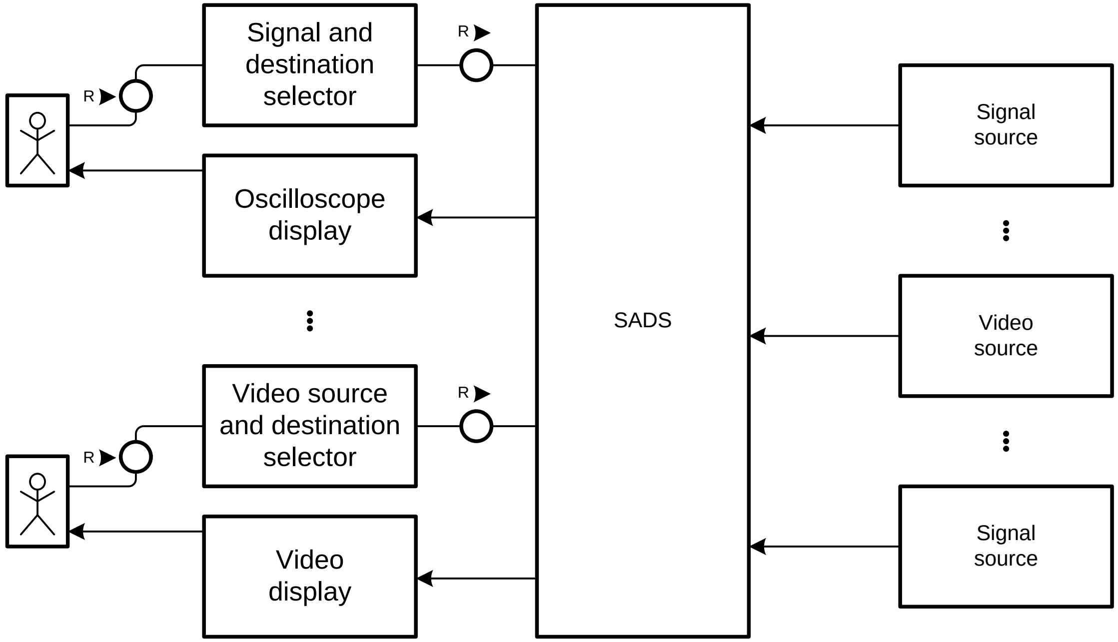
- Data structures that abstract from physical devices (DP)
- PVSS interacts with devices by reading and writing to elements (DPE) of data structures that represent those devices.
- Data point type (DPT) defines the structure of DP and contains a set of hierarchically ordered DPEs.
  - Can be of a simple type (e.g. string or integer) or a complex type (e.g. static or dynamic lists of integers).
- **Every subsystem** or component is **represented as a datapoint** in the system
- **All interaction** with components/subsystems goes **through** the **datapoints** to maintain consistency and operation safety

Accessing Device Components

- Supervisory panels to control entire subsystems
- Virtual Instrument panels
  - Communication via PVSS datapoints
  - Communication via NI SV

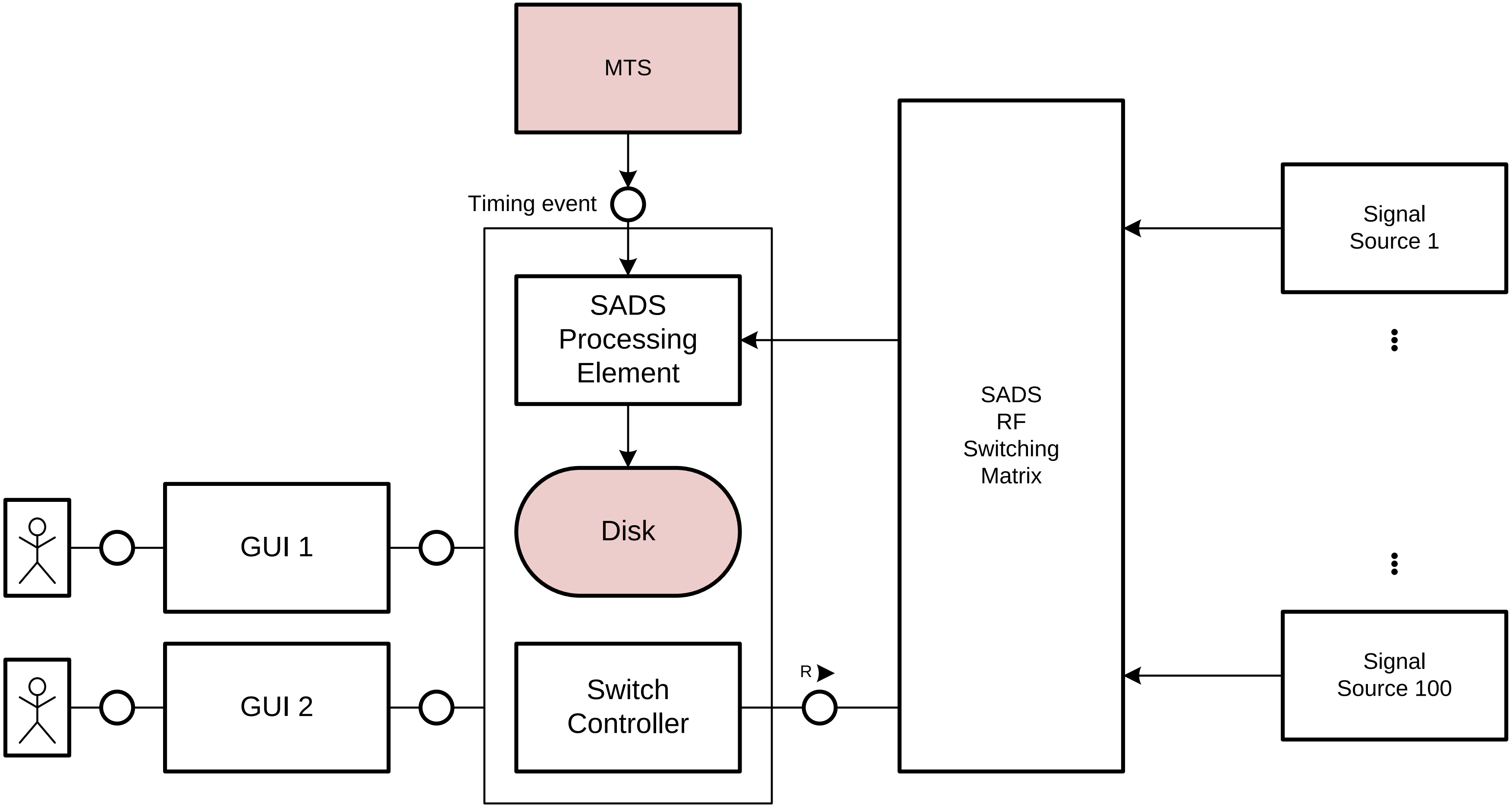


SADS



- Connect analogue signal sources to devices acting as sampling scopes display it on operator screens.
- Select video sources and tell on which screen to show them.
- Acts as MD, analysis tool not as primary operation display
- Requirements to be captured by October 2010

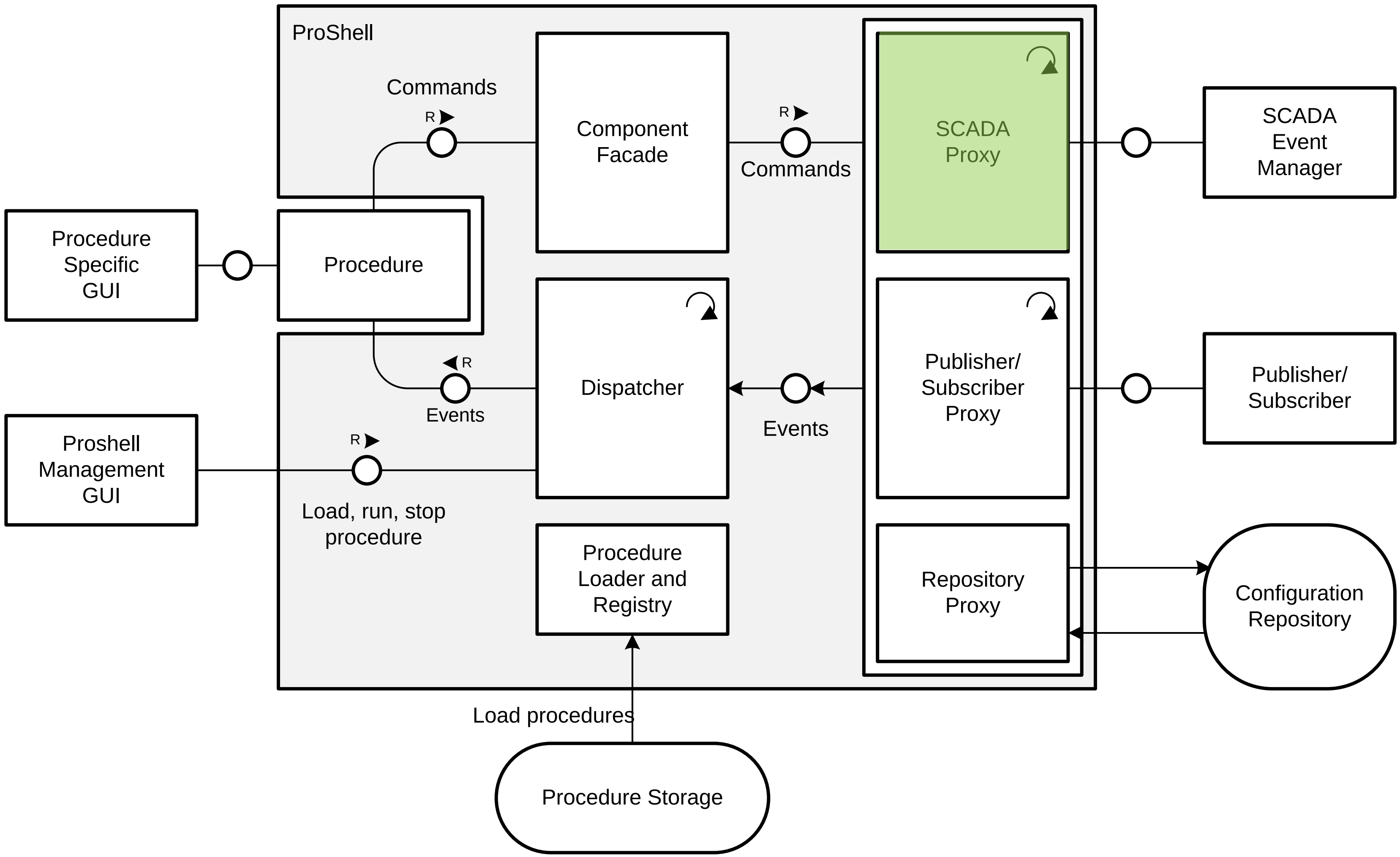
SADS Architecture



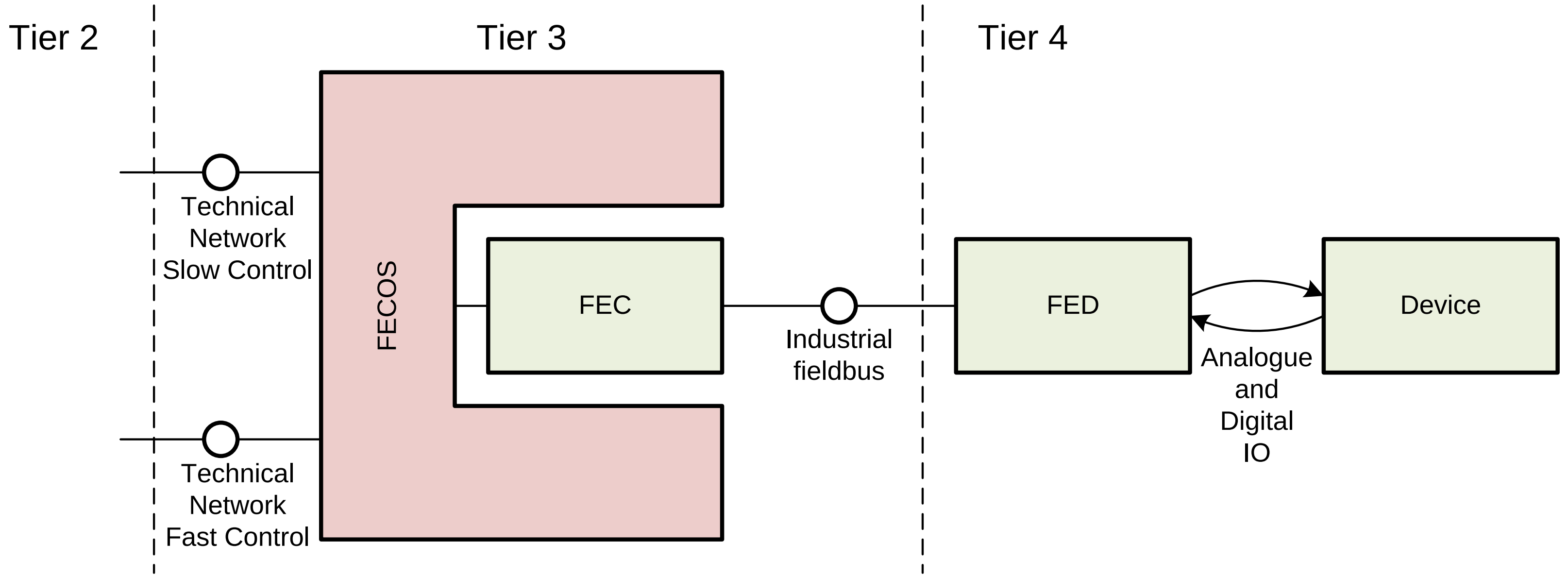
Procedure Framework (ProShell)

- Functionality **to automate repetitive supervisory tasks**
- Implement **beam diagnostics procedures**
- Framework written in-house in C#
- Procedures to be written by “experts” in C#
  - Training and documentation to be provided by WP CO.

ProShell Architecture



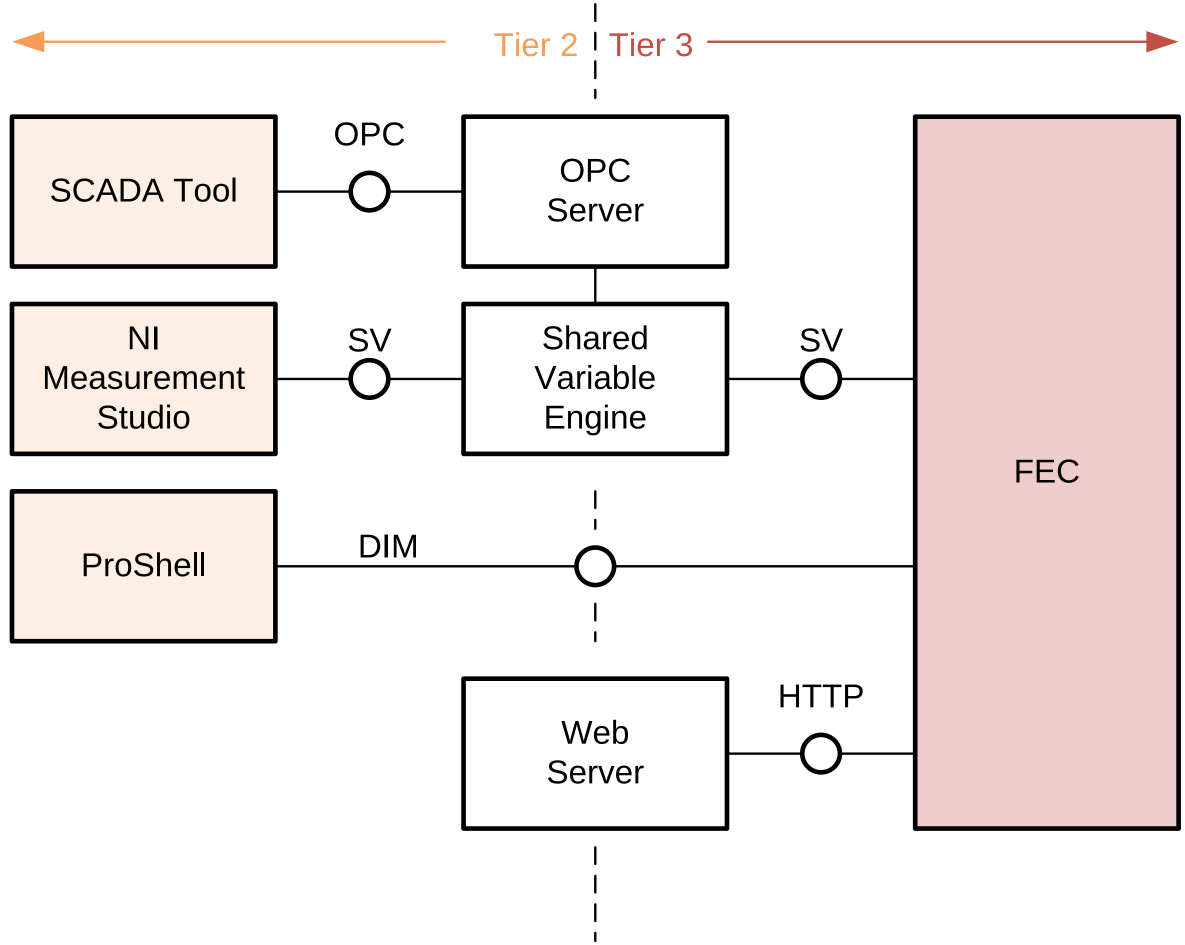
T3 and T4 Architecture



- WP CO provides LV-RT programming framework (**FECOS**)
- **Uniform state management, configuration, communication**
- FEC software components provided by WP CO for individual work packages on case-by-case basis (by default in WP)
- FEC can be operated standalone or via Working Set/VAcc

FEC Communication

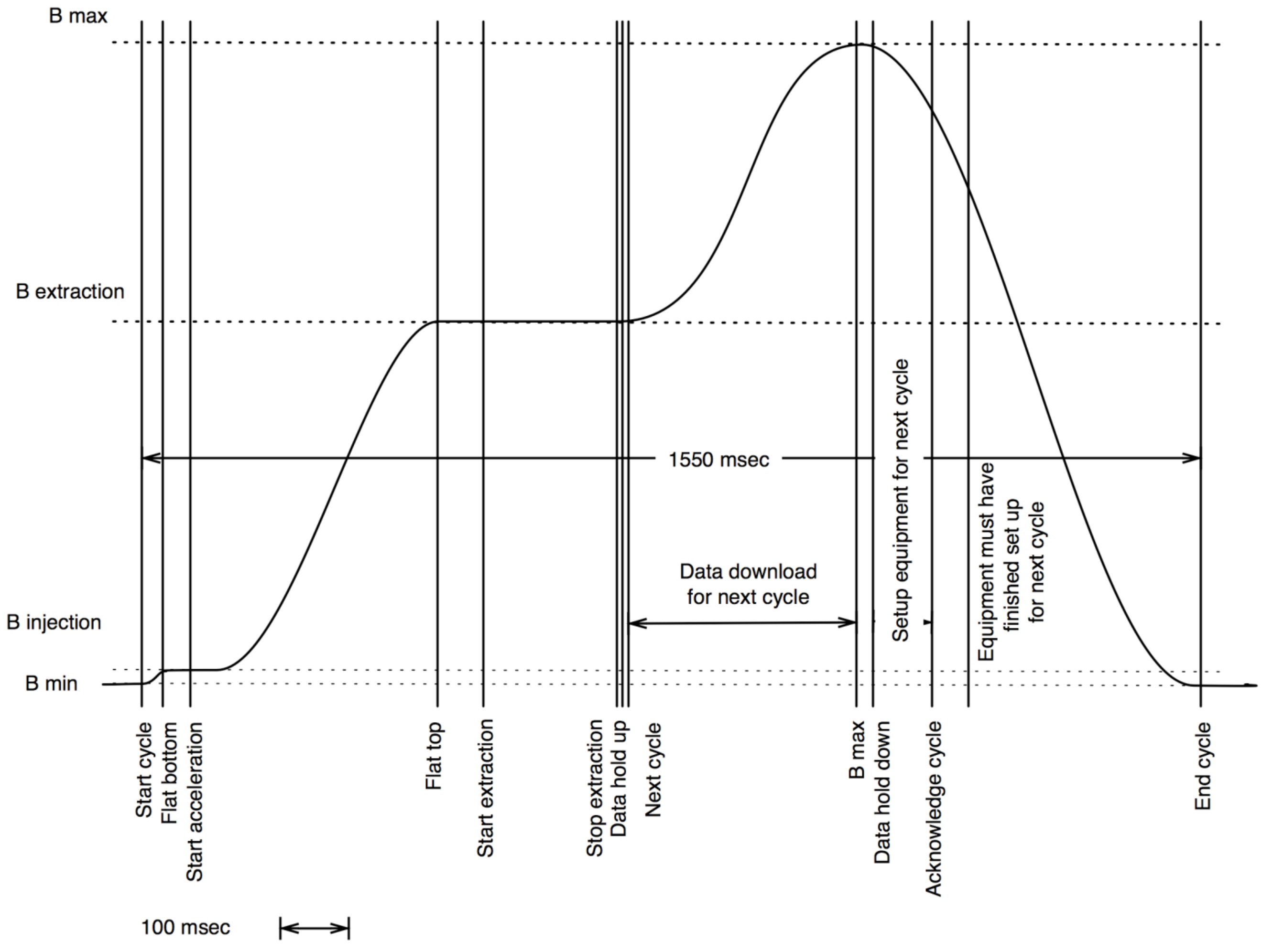
- FEC is building block at T3
- Defined interfaces to T2
- State machine and slow control via OPC
- Direct data access via NI Shared Variable engine and DIM
- BLOBs via HTTP



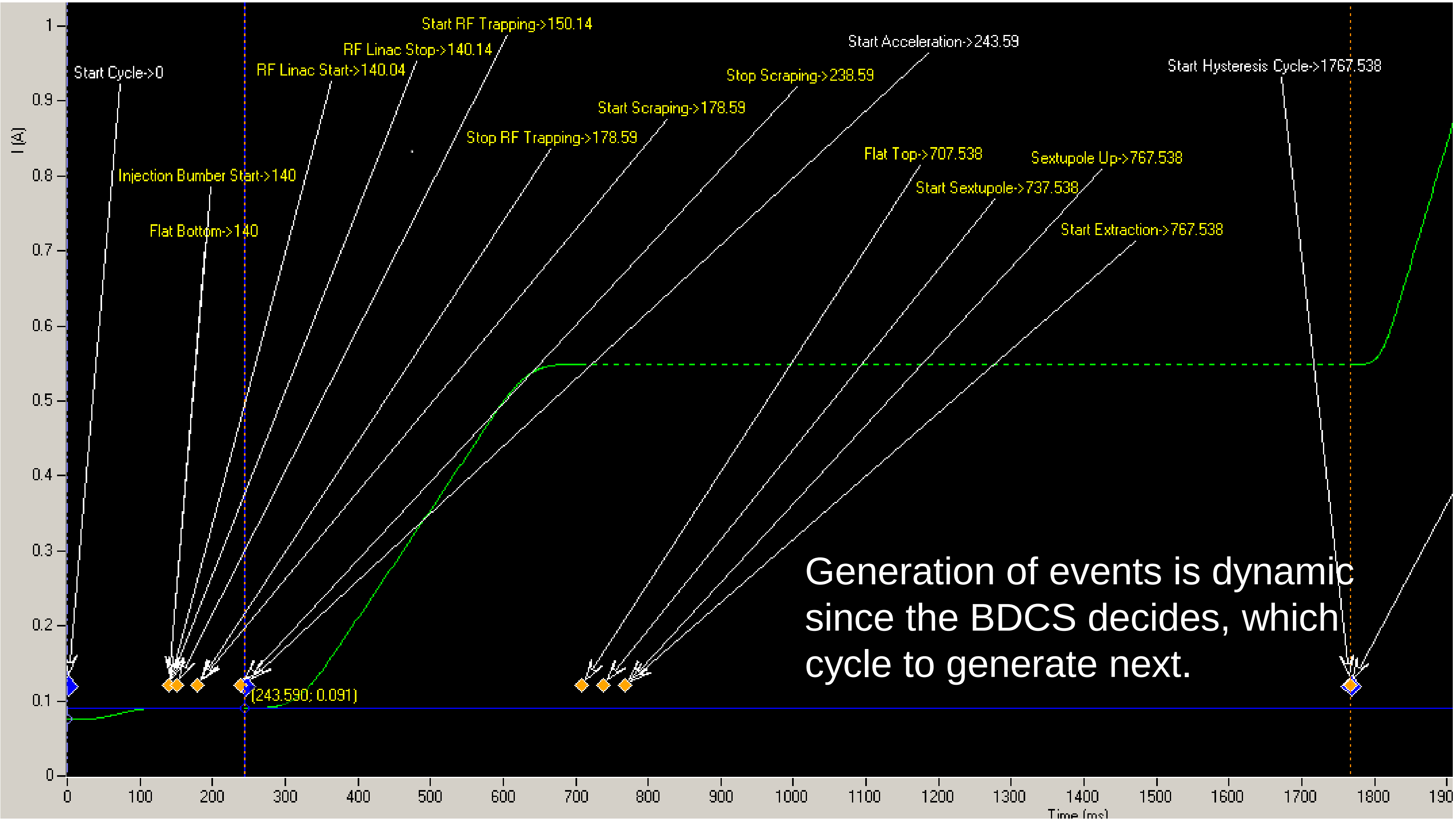
Main Timing System

- **Coordinates devices** in each subsystem (**ES-090512-a-JGU**)
  - Precision of ordinary timing events O (1 µsec)
  - Synchronization of 2 receivers O(50 nsec), local solution
  - Non real-time events distributed via DIM (Ethernet, TN)
  - Reference clock O(10 MHz) foreseen
  - Provision O(10 Hz) stability pulse that can be picked on demand
- Broadcasts events
  - Occurrences in time that are assigned a logical name
  - E.g. start injection, start acceleration, start extraction, etc.
  - Sequences are statically defined for individual cycles
- **Cycle definition** comprises
  - Record of settings (**WHAT**)
  - Sequence of timing events (**WHEN**)

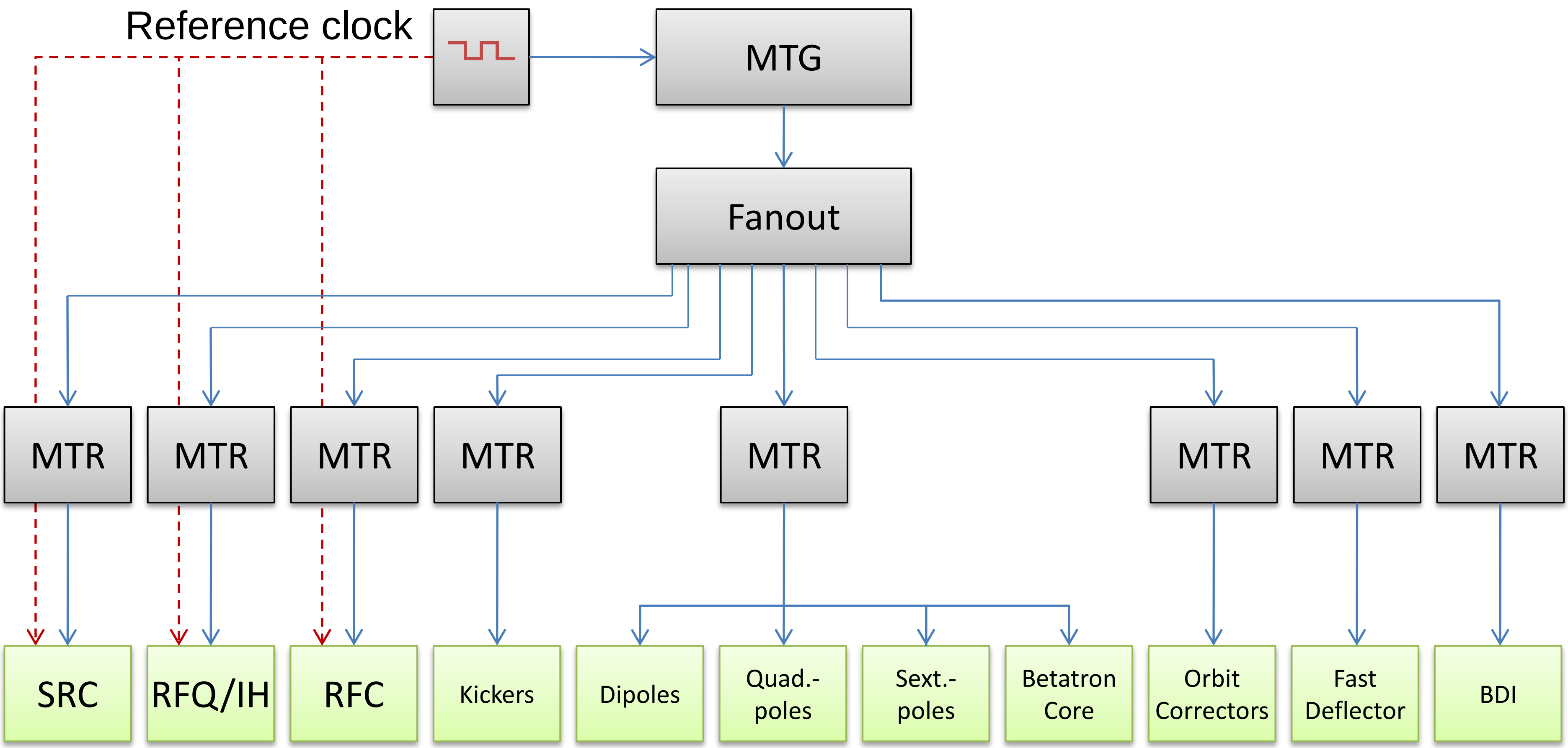
Typical Timing Events



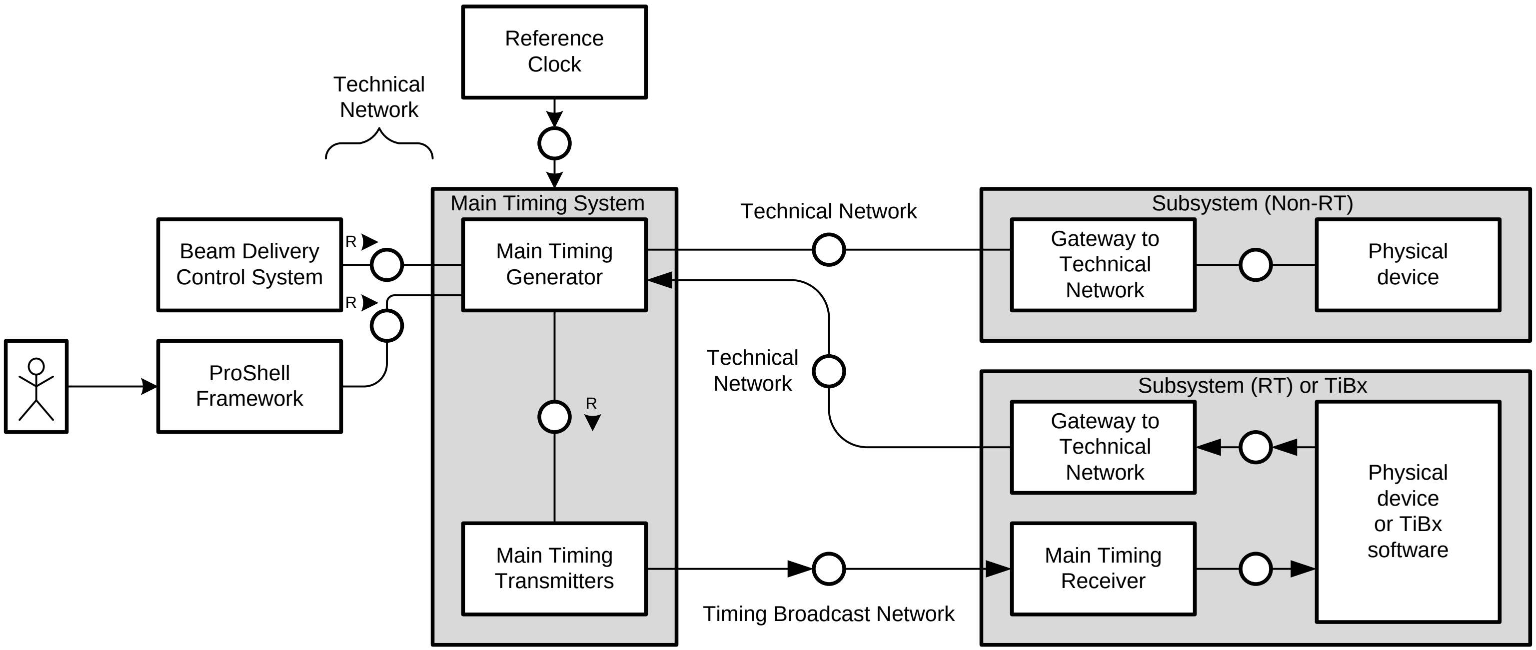
Definition of Timing Sequence



Main Timing System Concept

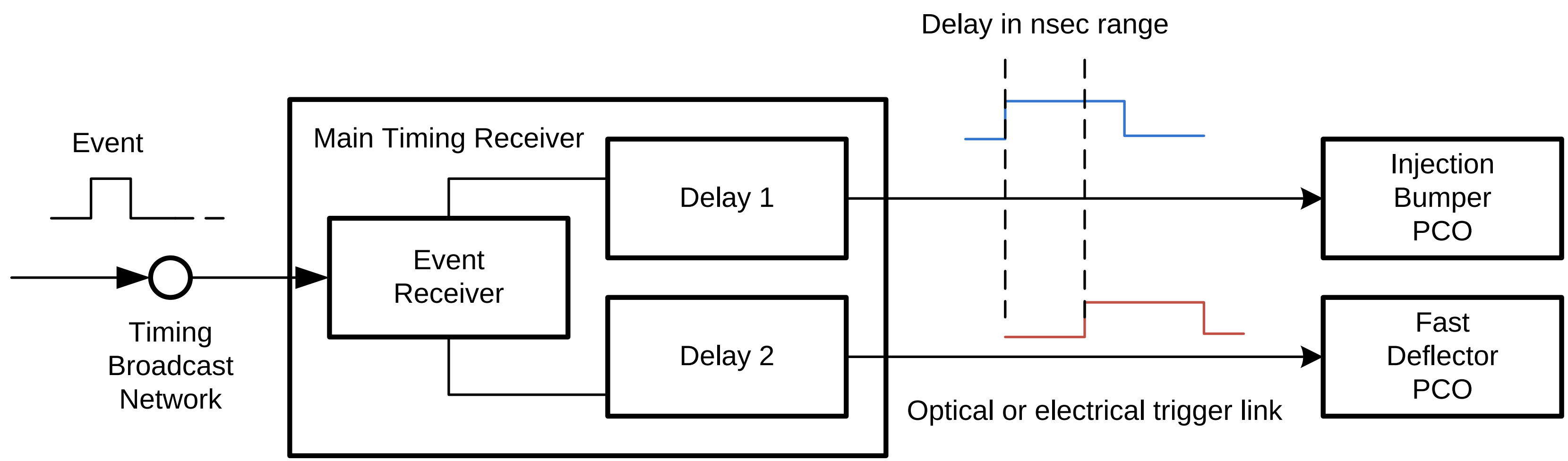


MTS Architecture



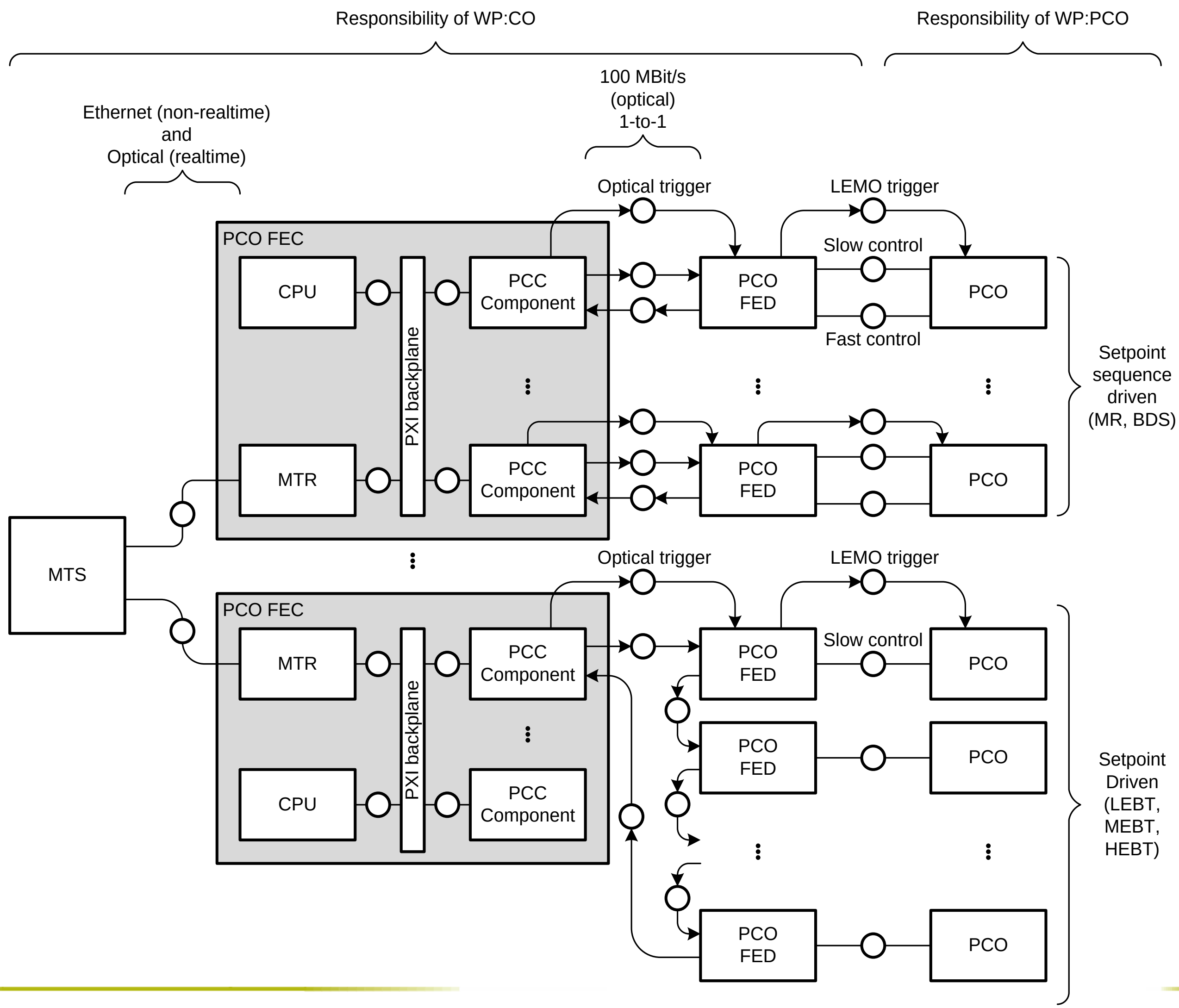
- Event broadcast sequencer (Main Timing Generator)
- **Real-time and non real-time** event broadcast
- **Timing Box** for scope trigger signal reception

High-Precision Operation

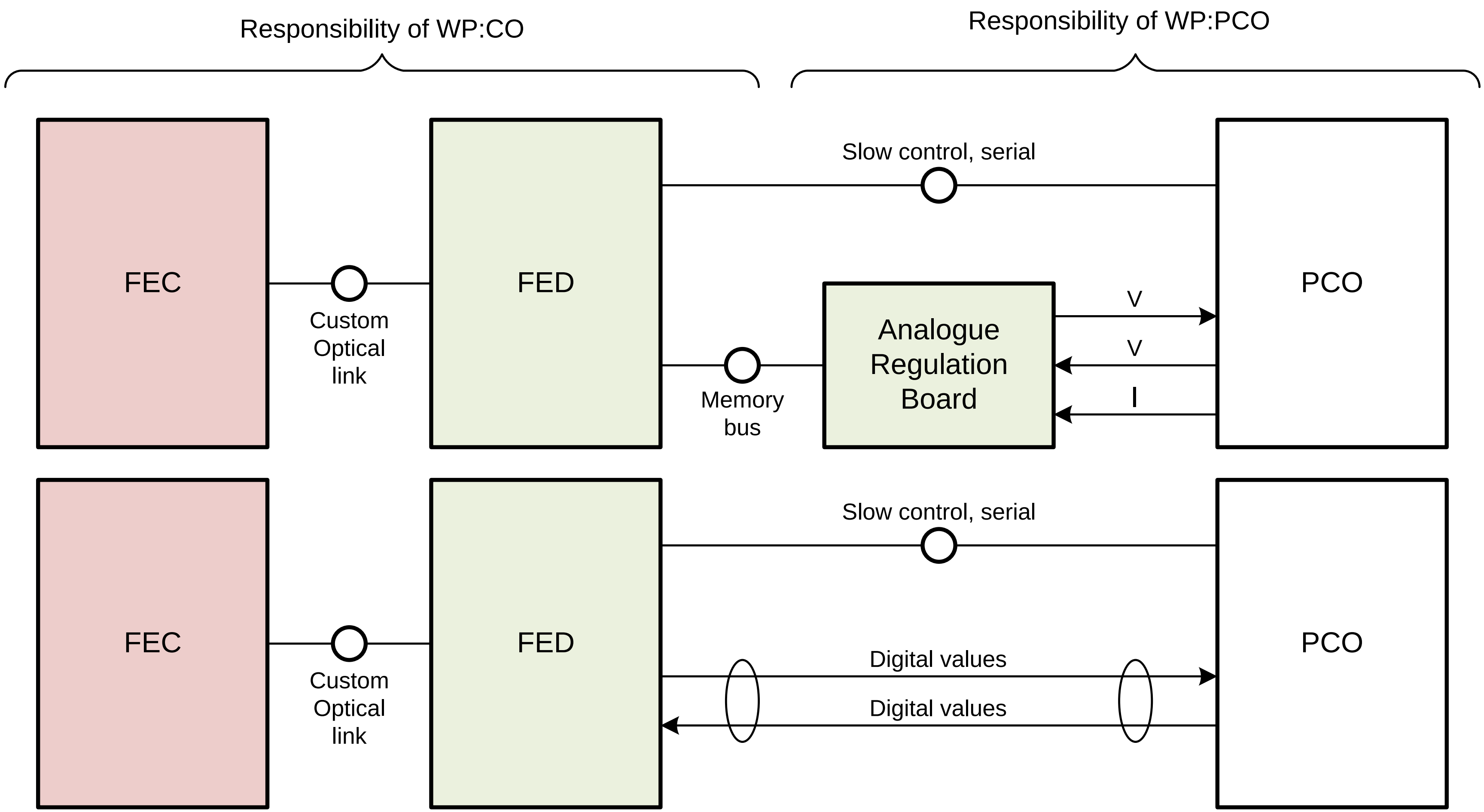


- Precision of MTS events 1 µsec
- For higher-precision, local solution based on programming the timing receiver board (MTR) possible
  - Dynamic reconfiguration not foreseen
  - Output can be simple optical or electrical signals
- Precision of O(nsecs) can be achieved

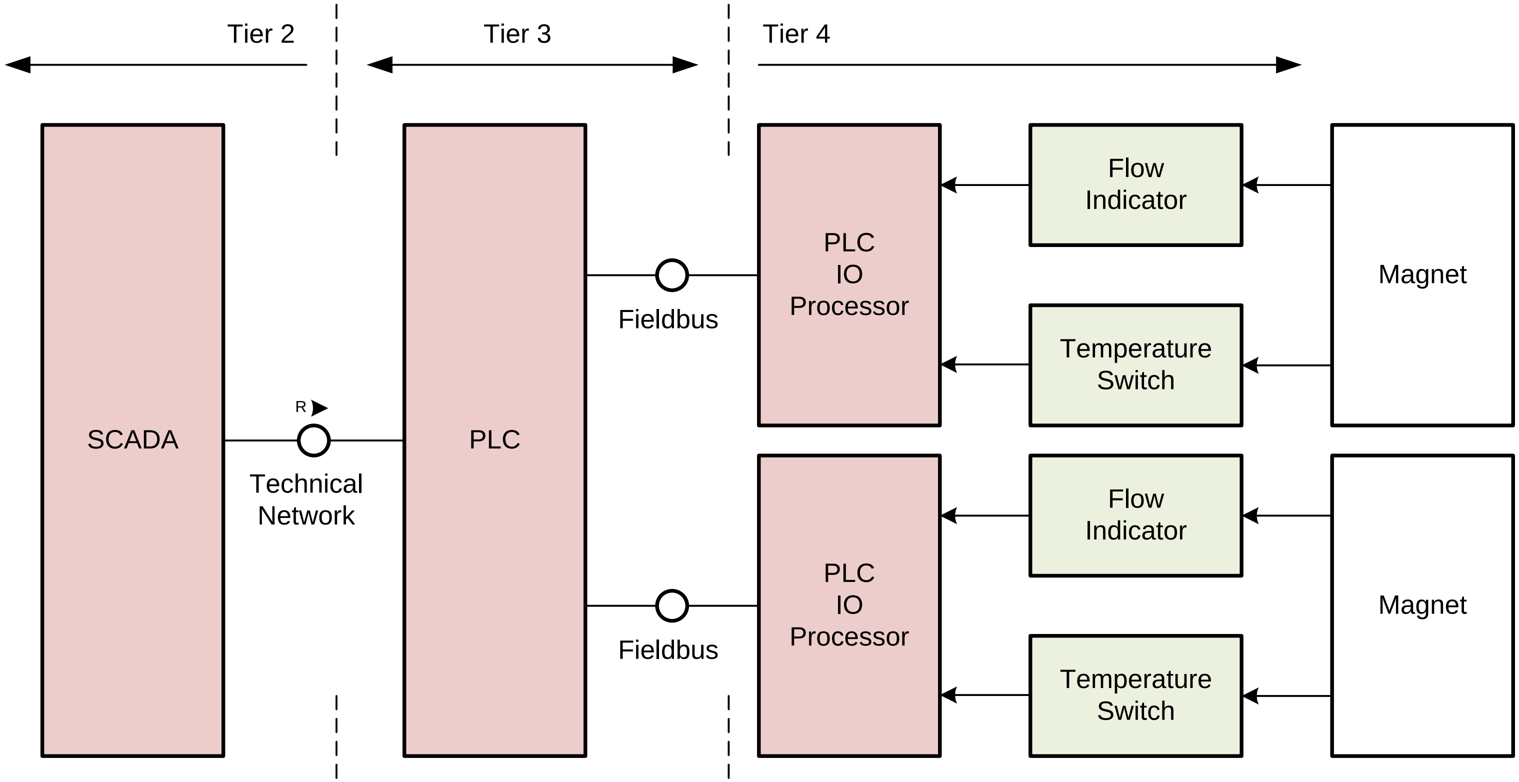
Power Converter Controller Architecture



PCC-FED-PCO Chain

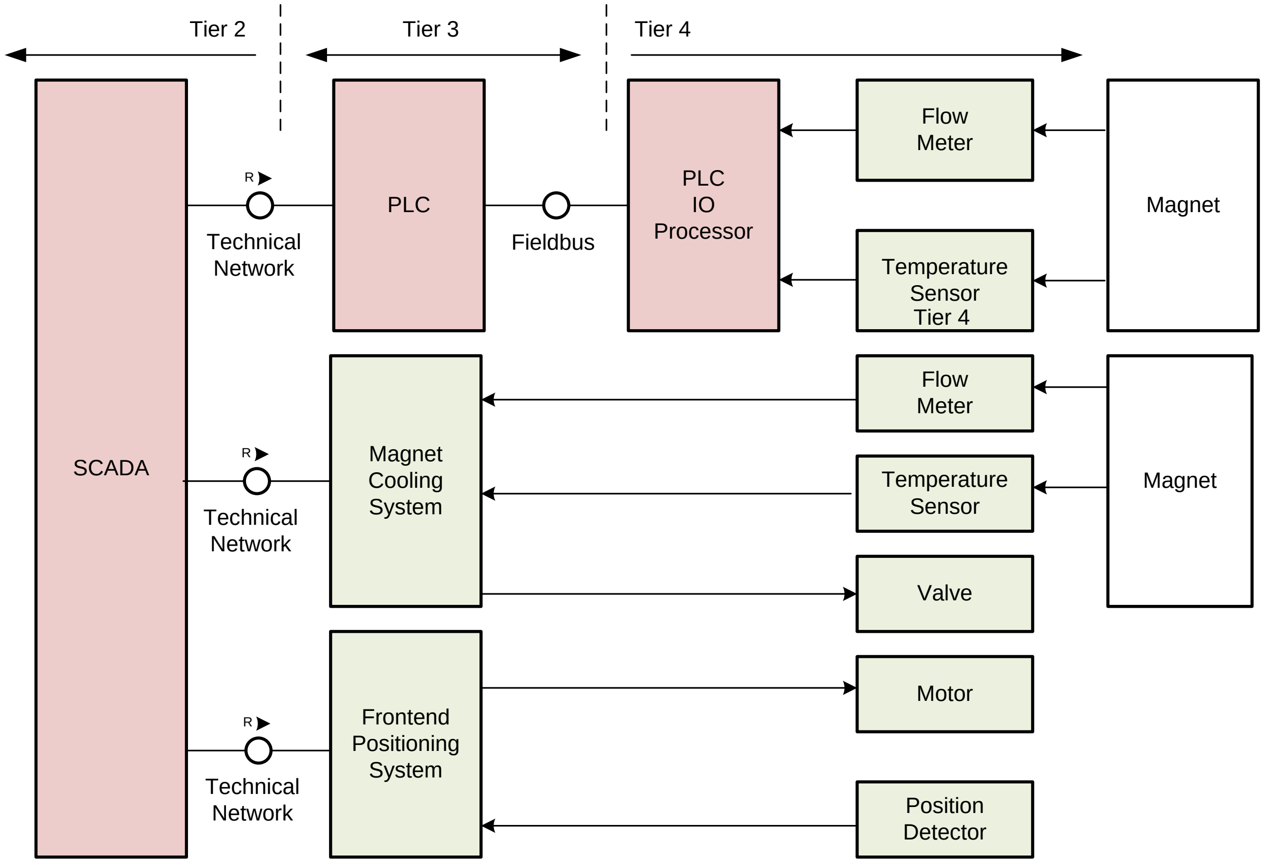


Conventional Magnets Slow Control



- Only Boolean values
- Will be implemented reading out Beam Interlock System (BIS)
- Saving on cabling via Profinet instead of point-to-point

Special Magnet Slow Control



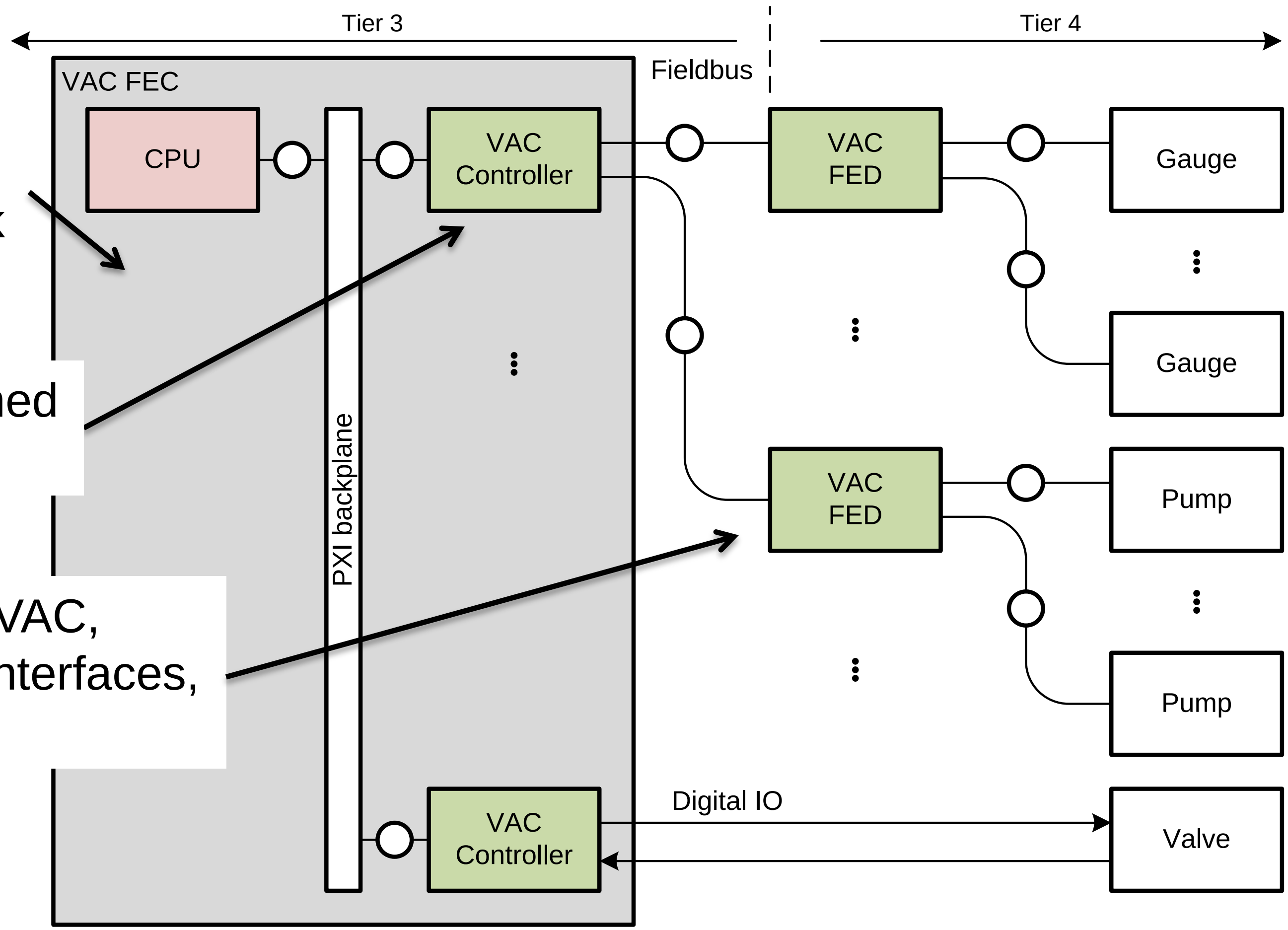
- Flow and Temperature via BIS
- Cooling and positionning subject to analysis (summer student)

Vacuum Control Architecture

FECOS and Crate Provided by WP CO, Mounted in same rack As FEDs

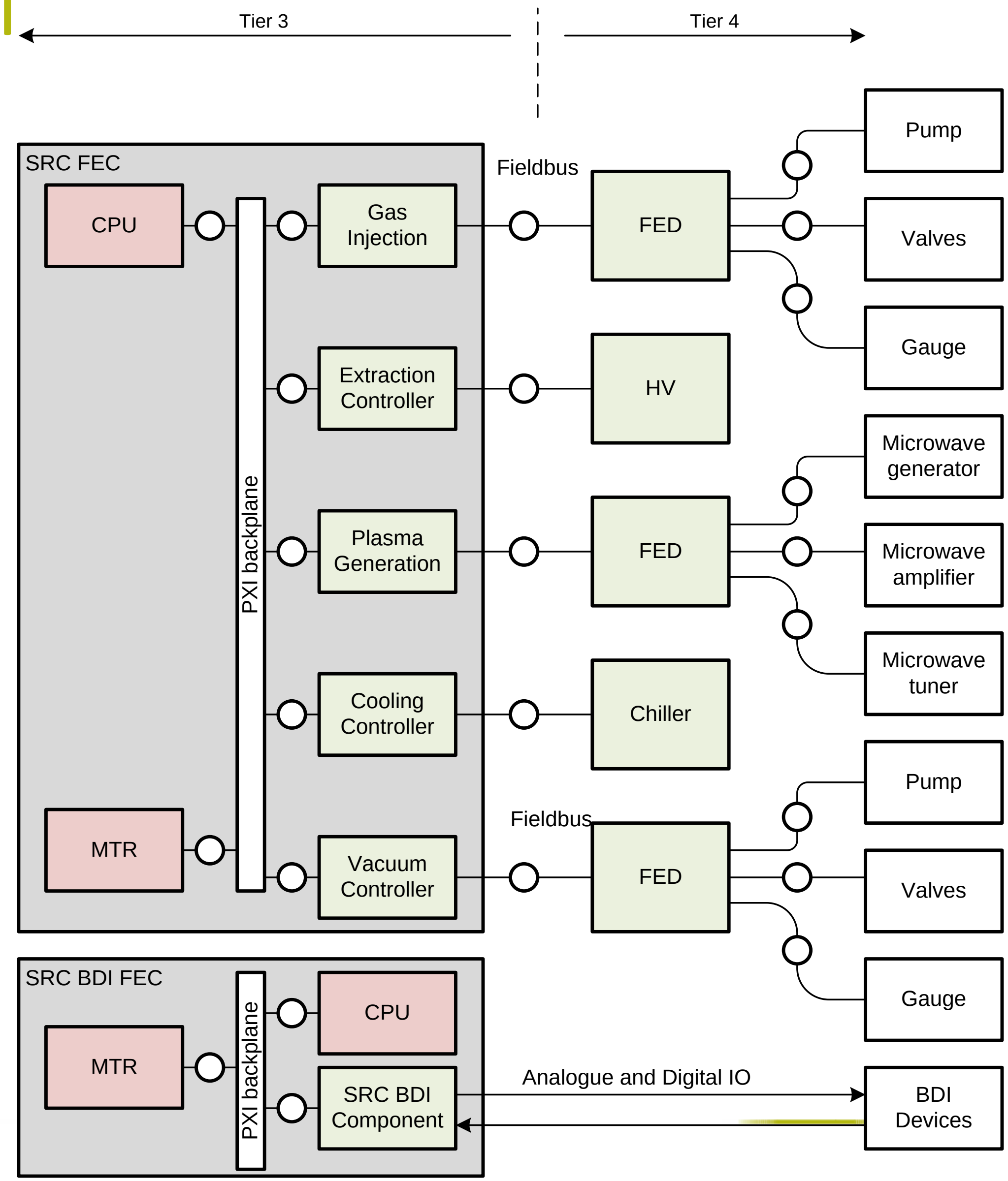
Component programmed By WP VAC

FED provided by WP VAC, serial RS422/RS485 interfaces, digital IO interfaces

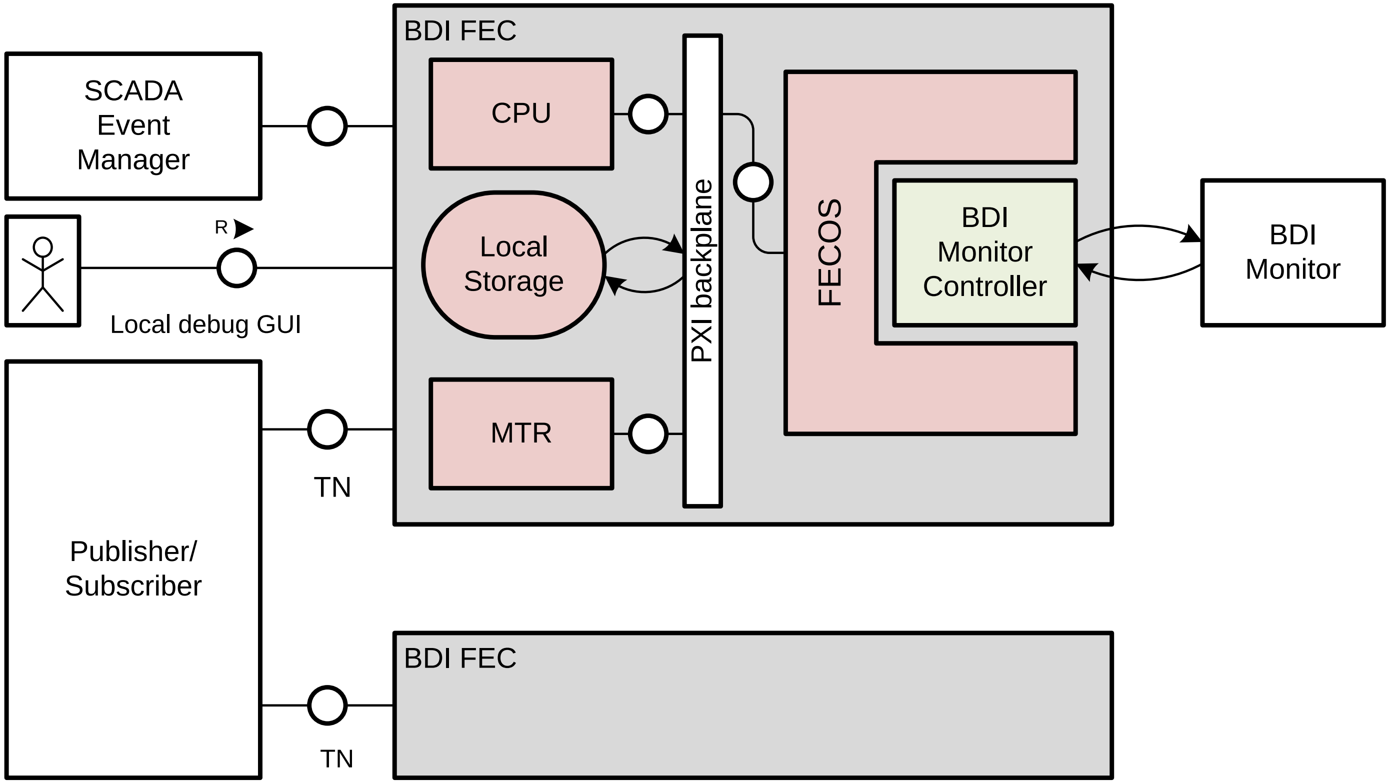


Sources Slow Control

- Existing LV software from supplier to be re-engineered
- CWO to Cosylab for provision of FECOS based component
- Beam diagnostics integrated into overall BDI system by BDI programmer
- Provision of MTS events foreseen
  - BDI, pulsed operation



Beam Diagnostics & Instrumentation

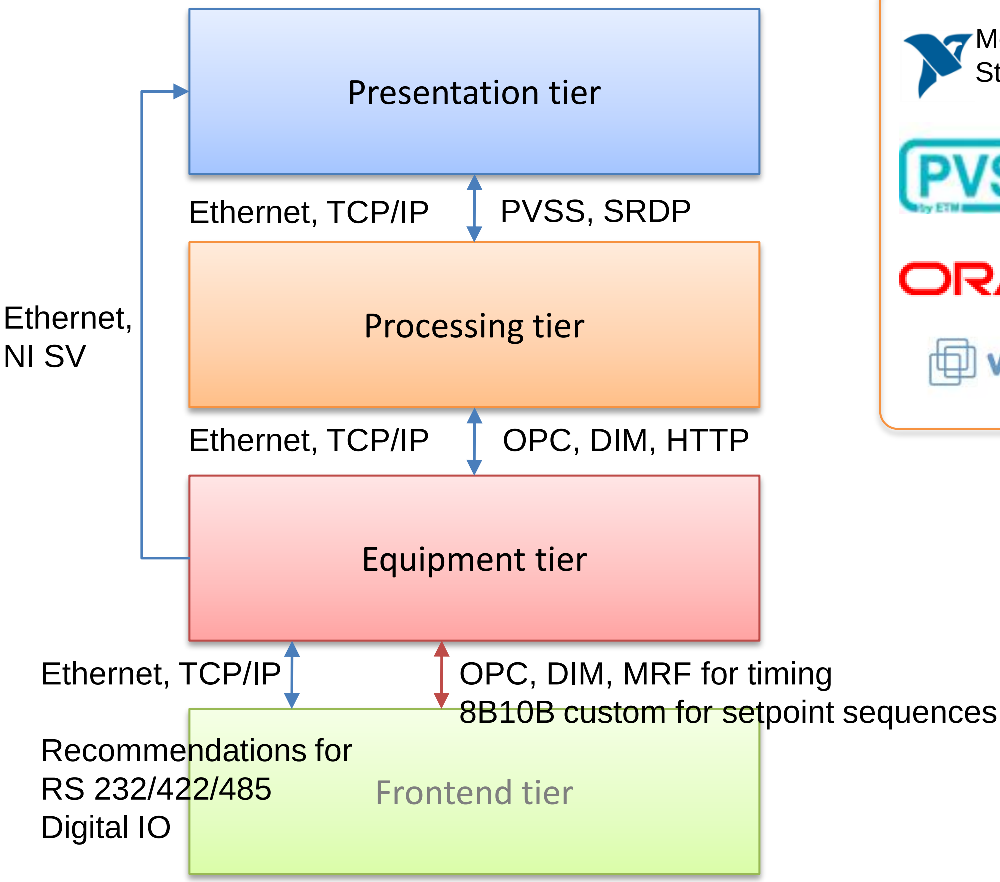


- WP BDI programmer works with FECOS provided by WP CO
- Standalone operation possible for each FEC
- GUI: ProShell and PVSS with Measurement Studio plugins

Technologies Used

- Primary means of communication with front-end:
  - Transport: Ethernet + TCP/IP, RS 232/422/485, digital I/O for interlocks
  - Protocols: OPC (+ NI SV) + DIM (+ HTTP)
  - Data contents: **Definition starts now based on CNAO XML schema**
  - PLCs: Siemens, S7 protocol over Ethernet preferred
- SCADA + GUIs: PVSS II + Measurement Studio
- Platform for Frontend:
  - PXI crates and crate controllers, FlexRIO FPGAs
  - LV-RT operating system
- Platform for T2,T3:
  - Rack-mounted server PCs
  - Virtualized MS Windows XP (upgrade to 7 for deployment)

Technology Overview



Measurement Studio

PVSS II

ORACLE

vmware

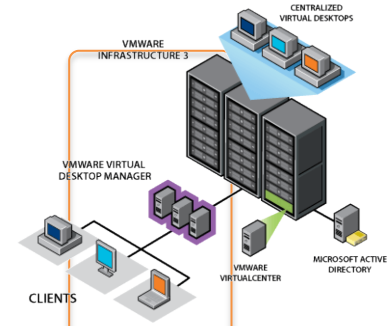
Inhouse C#

SIEMENS

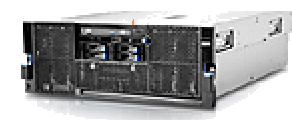
NATIONAL INSTRUMENTS

National Instruments LabVIEW

Virtual Machines



Rack mounted PC (PCIe)



PXI

PLC