FERMILAB-SLIDES-23-325-AD-PIP2

EPICS Deployment at Fermilab

Pierrick Hanlet

ICALEPCS 2023

2023 October 11

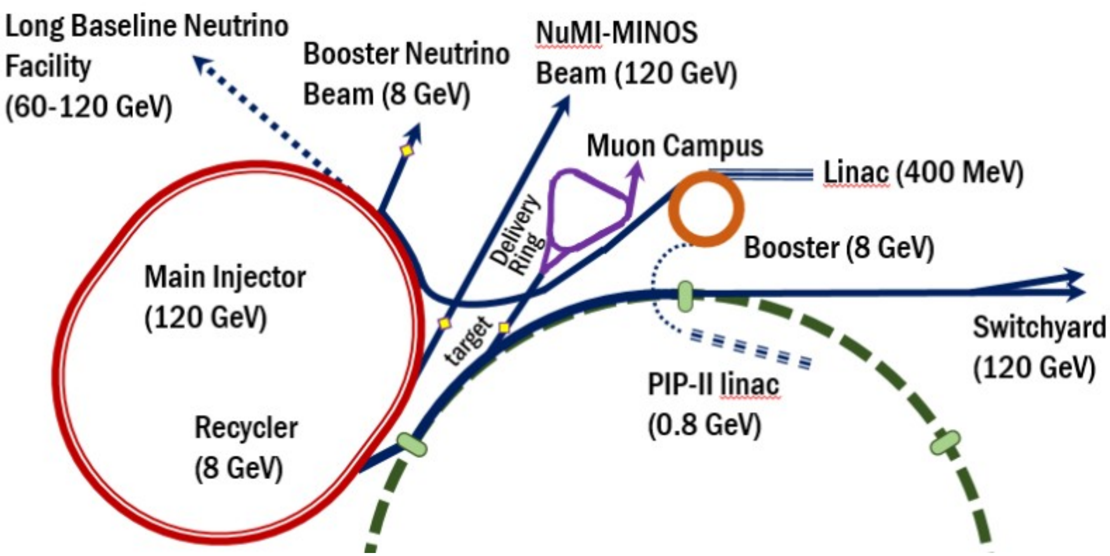
This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

A Partnership of:
- US/DOE
- India/DAE
- Italy/INFN
- UK/UKRI-STFC
- France/CEA, CNRS/IN2P3
- Poland/WUST

Content

- Introduction
  - Fermilab accelerator complex
  - PIP-II – see D. Nicklaus FR1BCO02
  - ACORN – see D. Finstrom TUMBCMO20
- Motivation
- Goals for Deployment
- CI/CD Pipeline
- Registering, Starting and Monitoring IOCs
- EPICS Experience at PIP2IT
- Summary

Fermilab Accelerator Complex



Series of accelerators
Provides beam to several experiments

- LBNF, DUNE
- NuMI
- muon campus
- local neutrino experiments
- test beams

PIP-II will replace the existing LINAC

PIP-II is the first US/DOE accelerator to be built with significant international contributions/partnerships.

PIP-II Mission

PIP-II is an essential upgrade to Fermilab accelerator complex to enable the world's most intense beam of neutrinos to LBNF/DUNE, and a broad physics research program for decades to come.

PIP-II Capabilities

Beam Power

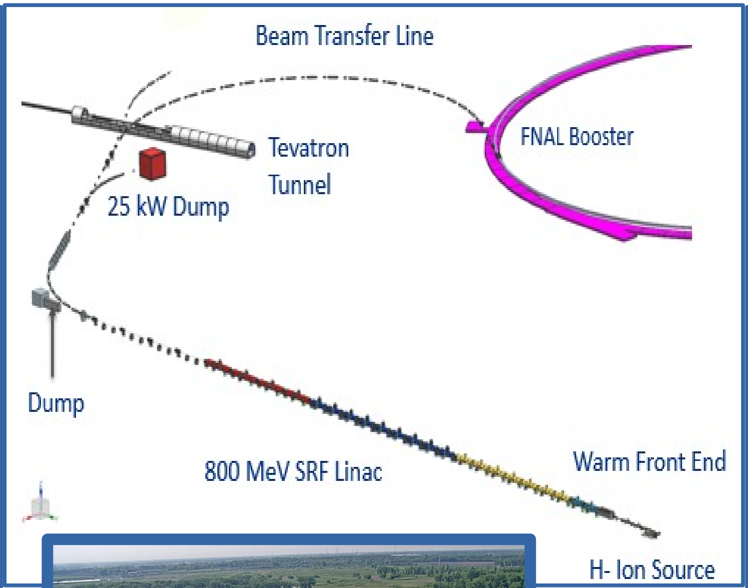
- 1.2 MW proton beam
- Upgradeable to multi-MW

Flexibility, multi-user capability

- CW-compatible
- Customized beams
- Multi-user delivery

Reliability

- Modernizes Fermilab accelerator complex



PIP-II Scope

800 MeV H^- SRF linac

- CW RF Operations

Linac → Booster transfer line

Accelerator Complex Upgrades

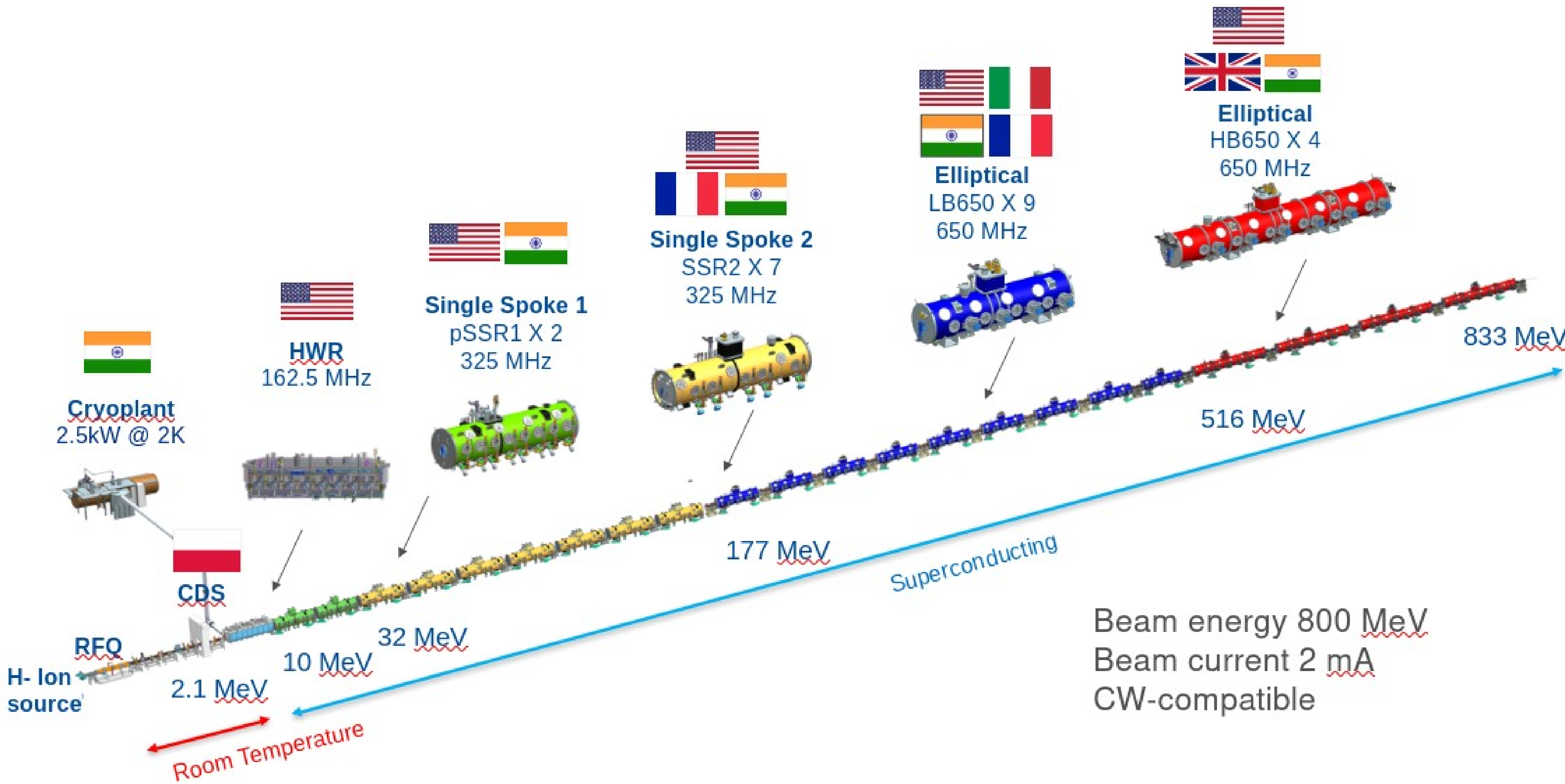
- Booster
- Recycler
- Main Injector

Conventional Facilities

- Space reserved for 2 CMs for 1GeV Upgrade

The PIP-II scope enables the accelerator complex to reach 1.2 MW p-beam on LBNF target

PIP-II Superconducting LINAC



→ For more on PIP-II see D. Nicklaus FR1BCO02

Motivation

- Treating EPICS deployment as a green field to simplify deployment for non-experts
  - https://ghe-pip2.fnal.gov/epics-controls/
- Small controls team, therefore we require:
  - robust build of infrastructure
  - automated build procedures
  - extensive testing
  - minimal functionality to automate deployment/monitoring of IOCs
- Developed a standard EPICS infrastructure to simplify developing IOCs for new developers
  - “base” and “Support” software are built (on all supported platforms) and made available on controls network
  - developers start from template IOCs and build against production ./base and ./Support
  - template IOCs have minimal basic functionality required of all FNAL IOCs
- Standard deployment and automated build for
  - robustness
  - ease in maintaining and debugging software
- Implement modern computing practices Continuous Integration/Continuous Deployment (CI/CD)
- Using PVXS protocol and disabling Channel Access (CA)
  - pvaccess protocol
  - structured data
  - already has ipV6
  - where network security measures will be implemented
- In kind contributions or commercial IOCs with CA will be accessible via p4p gateways

Goals for Deployment

“Treating EPICS deployment as a green field to simplify deployment for non-experts”

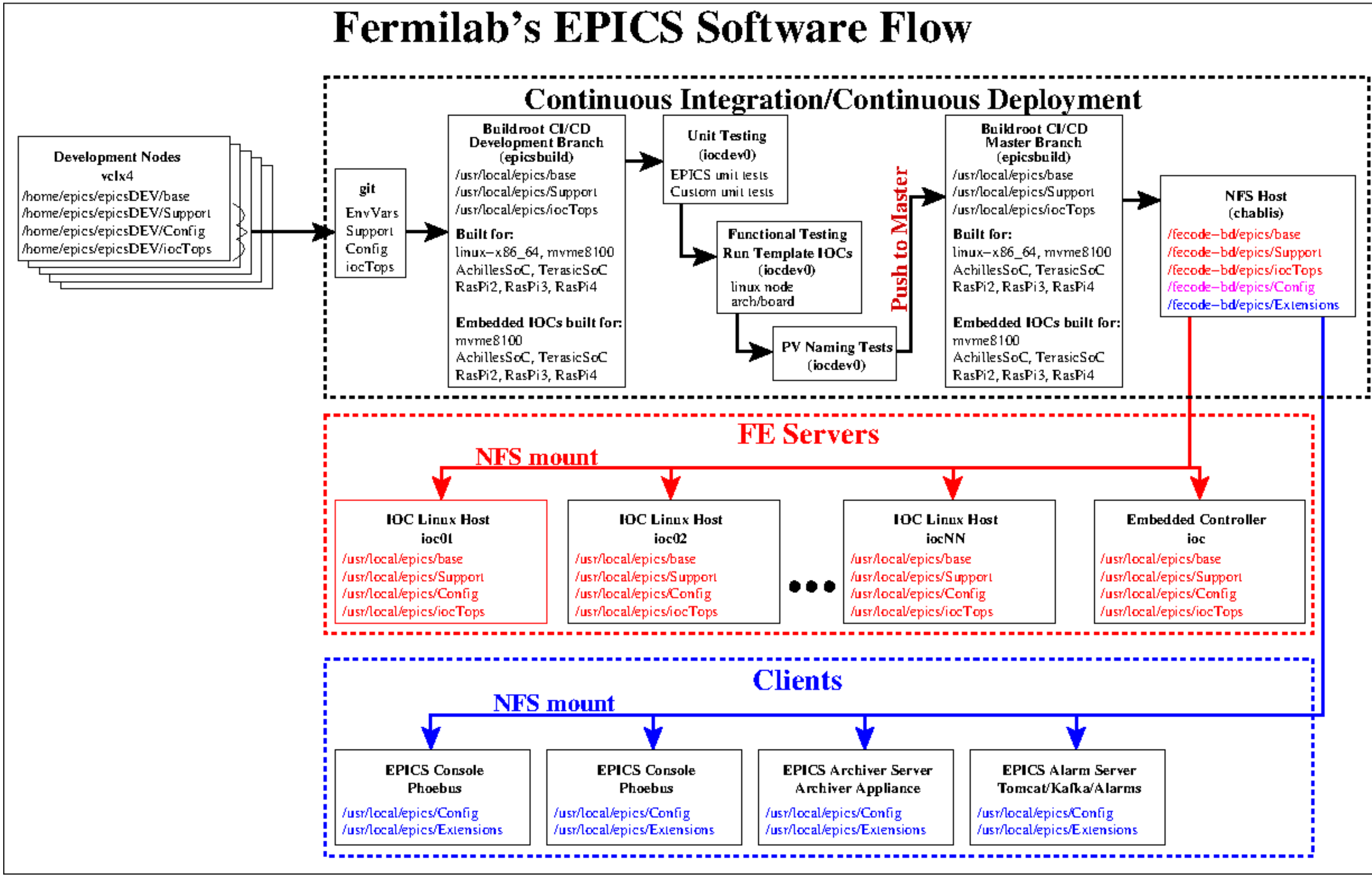
- PIP-II and possibly new components from ACORN will not rely on old hardware: VME, CAMAC
- Current EPICS versions work, and expected to continue to work, with the new hardware
- Leverage this to build current versions EPICS software in Continuous Integration/Continuous Deployment (CI/CD) pipeline and make all of EPICS base and EPICS Support/Modules “standard”
- Boiler plate code is made available on the Controls network from NFS mount: ***/usr/local/epics/***
- EPICS IOCs built against this boiler plate code base in CI/CD pipeline

In future, when present hardware becomes “old” hardware, we will use containers to preserve the IOCs

CI/CD Pipeline: Supporting Multiple Platforms

- In addition to standardizing code, we are exploring standardizing platforms:
- EPICS base, Support, & iocTops are built for different architectures/platforms:
  - linux-x86_64, arm/Cyclone-V, arm/Arria-10, arm/ZCU106 Xilinx, arm/RasPi2, arm/RasPi3 arm/RasPi4
- Using Buildroot for raspberry pi builds and arm/Cyclone-V and arm/Arria-10
- Adding Yacto for Xilinx
- These produce kernels and root file systems for each platform
  - build process copies these to tftp area
  - required for network booting embedded nodes
- Also generates Toolchains or SDKs for cross compiling
- Toolchains/SDKs used to build EPICS code base
- CI/CD artifacts are deployed on NFS host and available on NFS mounts:
  - /usr/local/epics/base/bin/linux-x86_64
  - /usr/local/epics/base/bin/linux-arm_raspberrypi2
  - /usr/local/epics/base/bin/linux-arm_raspberrypi3
  - /usr/local/epics/base/bin/linux-arm_raspberrypi4
  - /usr/local/epics/base/bin/linux-arm_terasicsoc
  - /usr/local/epics/base/bin/linux-arm_achilles
  - /usr/local/epics/base/bin/linux-arm_zcu106
- Same for /usr/local/epics/base/lib, /usr/local/epics/Support/xxxMODULE/lib, etc

CI/CD – Software Path to Deployment



CI/CD Pipeline

- Implementation developed by Mariana González.
- Using Github for code management, documentation and issue tracking.
- Using Github Actions tool to automate building and testing
- Full CI/CD chain is complete and successfully tested
  - Code and documentation migrated Github
  - Automated build of the 3-tier EPICS for all supported architectures
  - Automated unit testing – working for host architecture
  - Automated testing:
    - Unit tests
    - Basic functional tests (based on templateIOC tests)
    - Check for duplicate PV names
- Passing tests allows for code to be build, tagged, and pushed to NFS host

- IOC owner must register IOC before deploying
- Presently building base-7.0.7 on Alma Linux 9.2

CI/CD Pipeline: Template IOCs

Each Fermilab IOC running on the controls network will provide the following:

- heartbeat
- IOC statistics (CPU usage, memory usage, etc.)
- capability of 20 Hz scan rate
- use aSub (specific Support module) record for interfacing with IOC specific custom code (C/C++ libraries)
- access to Acnet (acnetPV wrapper)
- tcast – interface to clock system
- reccaster – Channel Finder

It is recommended to use the template in the following steps:

- build and run the IOC as is to test operability of your soft or embedded IOC
- once established, create IOC specific PVs, code, and link to IOC specific custom libraries

CI/CD Pipeline: IOC Production Deployment

We have 2 classes of deployment:

- Soft IOCs
  - all soft IOCs are hosted on linux servers
  - Alma Linux 9.2 – presently
  - all code is based on CI/CD output, so no need for containers
  - launched via procServ on different TCP ports
  - procServ scripts launched from *systemd*
  - log files in **/scratch/epicsLogs**

- Embedded IOCs
  - Raspberry Pis, SoMs (Achilles SoC, TerasicSoC, Xilinx)
    - Raspberry Pi will be single process server
  - SD card or on board flash will have host specific network configuration
  - kernels, root file system, dtb files, etc. from tftp server for network boot
  - use u-boot
  - NFS mount **/usr/local/epics**
  - IOCs launched via procServ and launched from *systemd*
  - log files in **/scratch/epicsLogs** which is mounted from soft IOC server(s)

Services

In addition to the infrastructure for IOCs, Fermilab is evaluating several EPICS services.

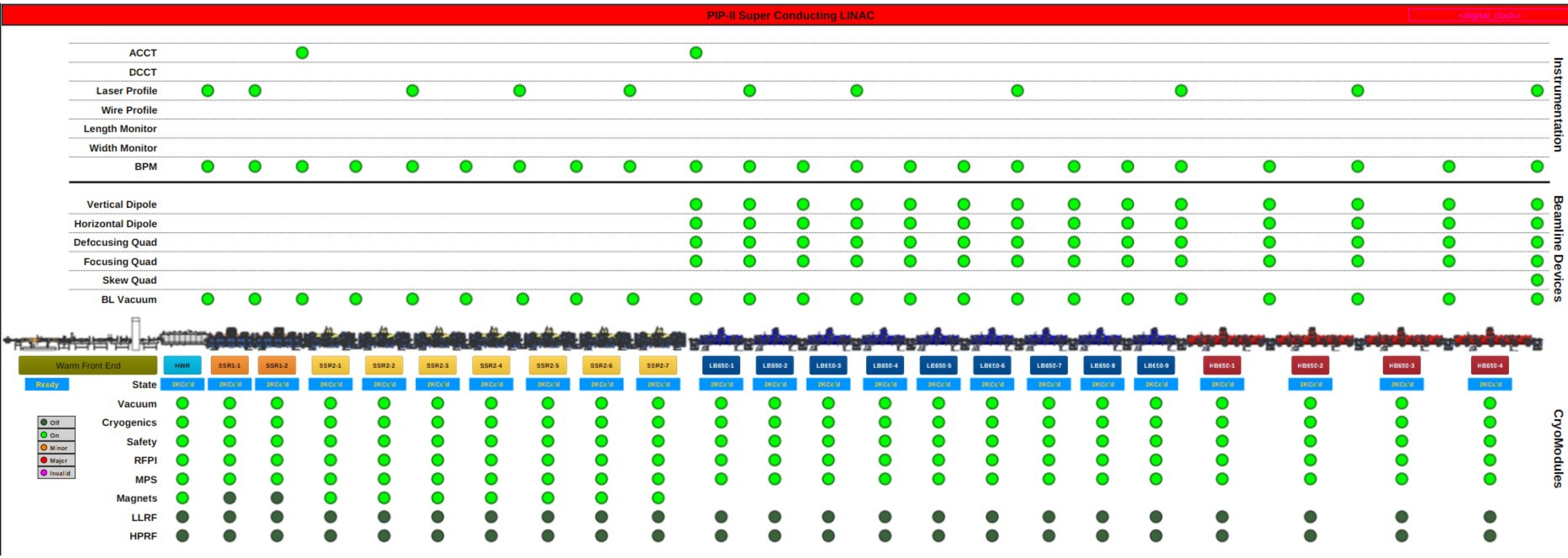
- The EPICS paradigm is (typically) to have all of the smarts built into the IOCs:
  - Calibrations
  - Sequences
  - Alarm limits and severity for each alarm
- Each service/user application has all information available to it without further processing
- Services include:
  - Consoles
  - Archiver
  - Alarm Handler
  - Channel Finder
  - Save & Restore
- These services have been installed and operated at PIP2IT

Services – Consoles

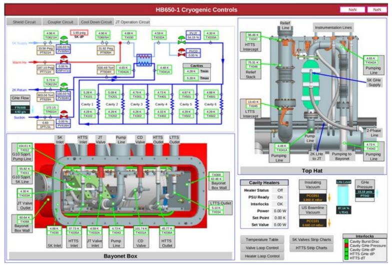
- Consoles refer to the operator visual screens (GUIs or HMIs) which will be used to monitor and control PIP-II apparatus
- Phoebus has been chosen as our principle platform (for now)
  - Based on Control System Studio (CSS), but rewritten without Eclipse backbone
  - Built in hooks for easy integration with different central services
  - Easy to use
  - Mature, flexible, and well documented
  - Drag and drop capability for developing GUIs
  - Can be used to launch physics applications and sequences

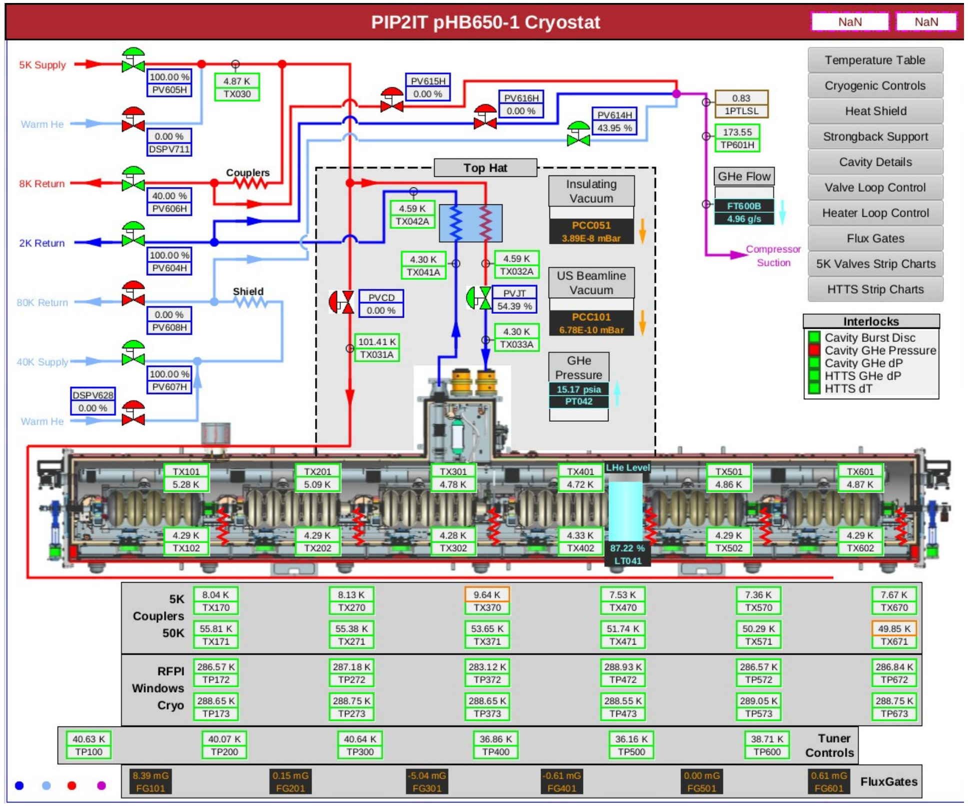
Services – Consoles

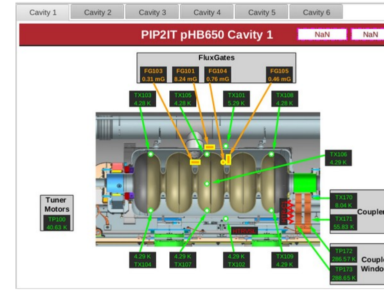
- Inspired by SLAC main HMI
- Status at a glance
- Launcher

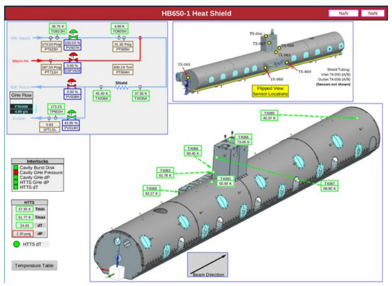


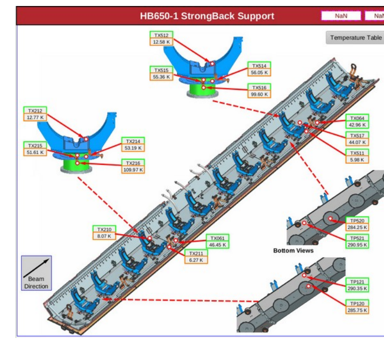
Services – Consoles

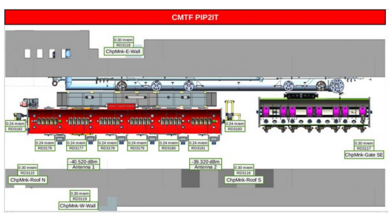


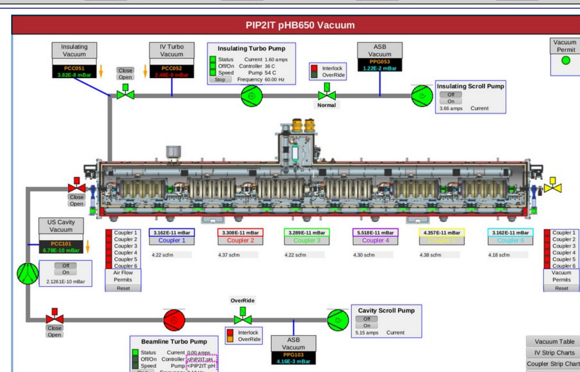


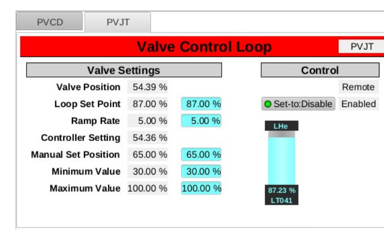


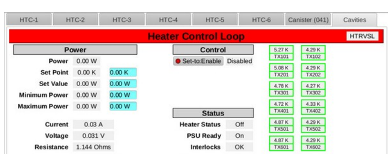












PIP-II

PIP2IT Lessons Learned

- PIP2IT is the CM test stand
- First CM successfully tested June 2023
- Also test stand for EPICS IOCs and services
- First use of deployment of EPICS infrastructure
- IOC deployment worked well
- Big Lesson: cannot install & deploy services and forget them
  - Phoebus worked well, but slow when hosted from linux servers to windows
  - bizarre java errors
  - GUI updates
  - need proper configuration of Phoebus alarms, Archiver Appliance, Channel Finder, cannot simply install and ignore

Concluding Remarks

- EPICS is now well established at Fermilab and is here to stay
- Green field allows us to build with latest software versions
- Our CI/CD pipeline is fully functional
- Many tests still to write
- Network booting embedded systems and NFS mounted EPICS code (IOCs)
- Deploying several EPICS services (Consoles, Alarms, Archiver Appliance, Save & Restore, and Channel Finder)
- Services need as much (perhaps more) support than IOCs

PIP2IT – Lessons Learned

- **At PIP2IT:**
  - Phoebus is running on all consoles at Cryomodule Test Facility (CMTF)
  - LLRF is still using edm for HMIs – will be transitioned to Phoebus
  - What we hope to learn:
    - remote access for production
    - performance capabilities
    - other (non-Phoebus) application platforms
    - consider dedicated linux servers – one server/console
    - main control room integration
    - study authentication and authorization
    - testing other applications: archiver, alarms, channel finder, save & restore