OAK RIDGE National Laboratory

Integrated Control System Architecture for STS Project

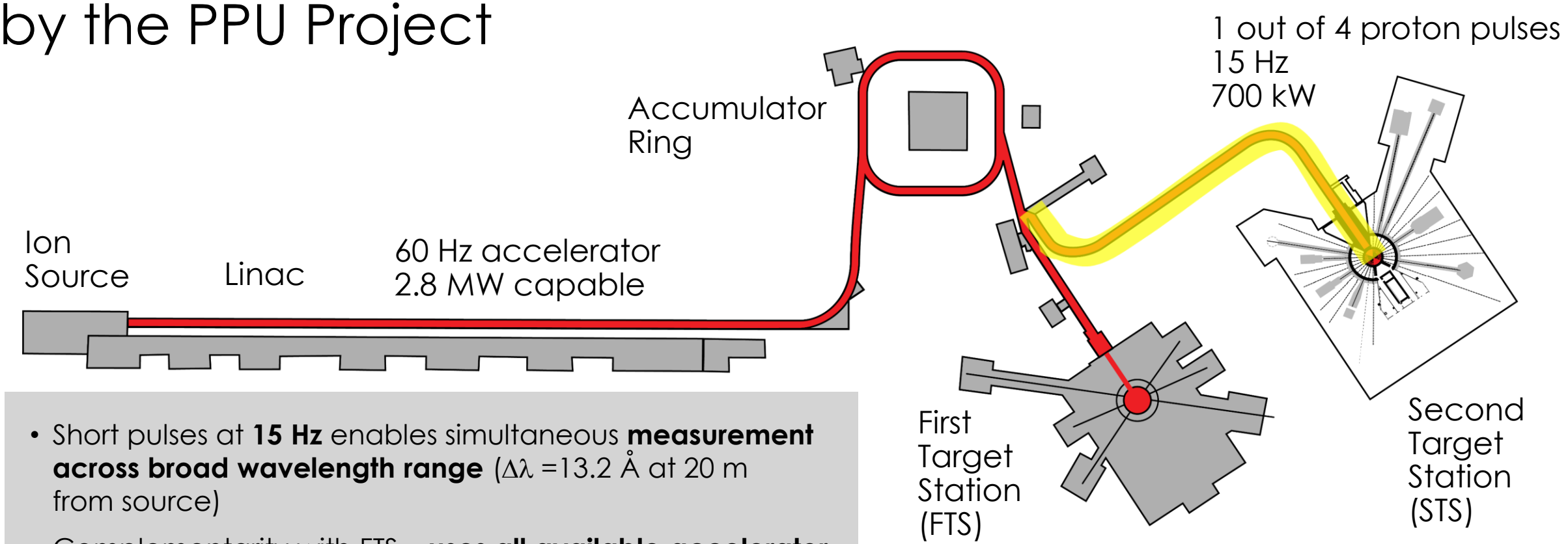
Jay Yan, Steven Hartman, Kay Kasemir, Matthew Pearson
STS Integrated Control Systems
Oak Ridge National Laboratory

EPICS Collaboration Meeting, Cape Town, South Africa

Oct. 8, 2023

U.S. DEPARTMENT OF ENERGY

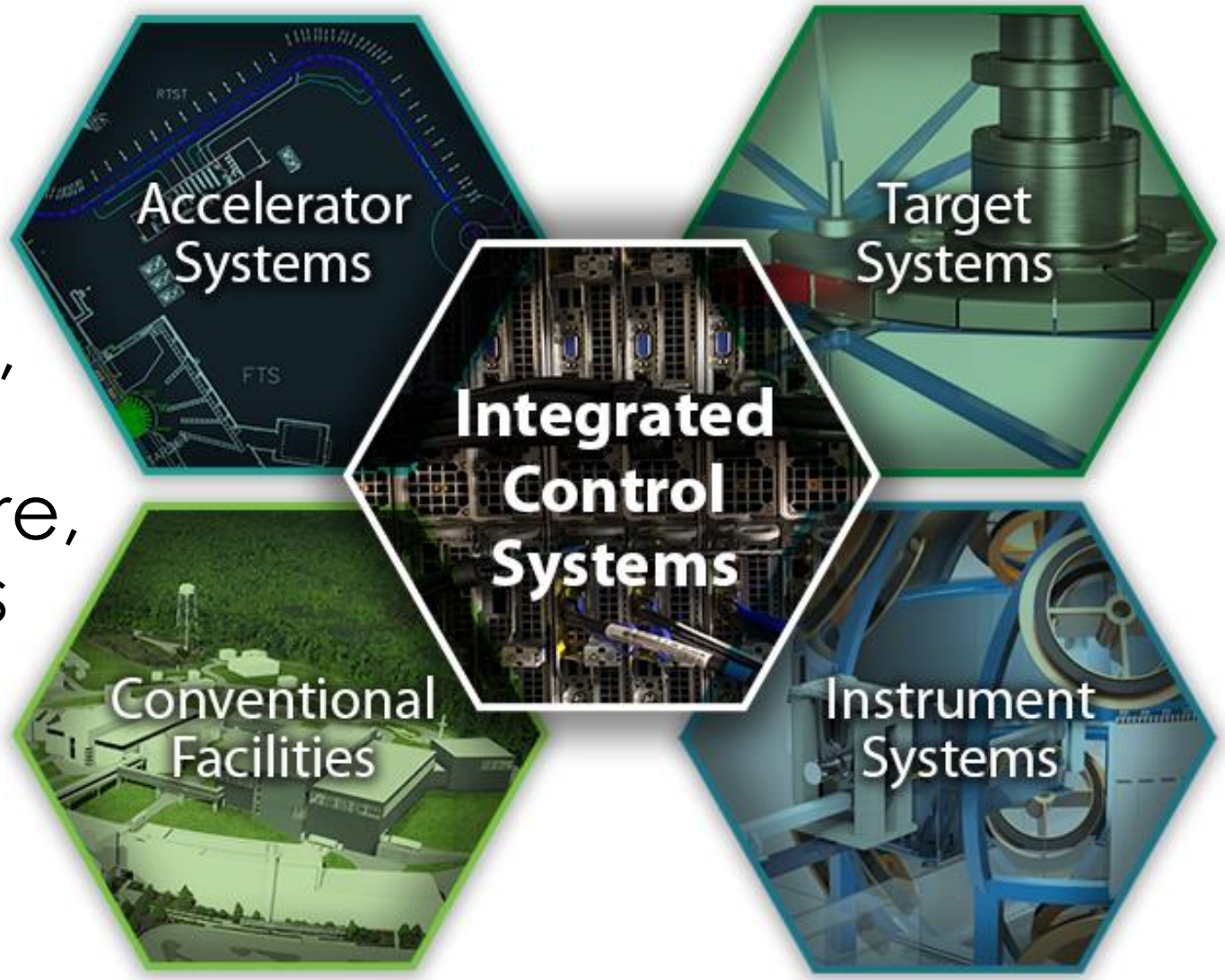
STS leverages the existing accelerator infrastructure of the SNS and utilizes the additional beam power provided by the PPU Project



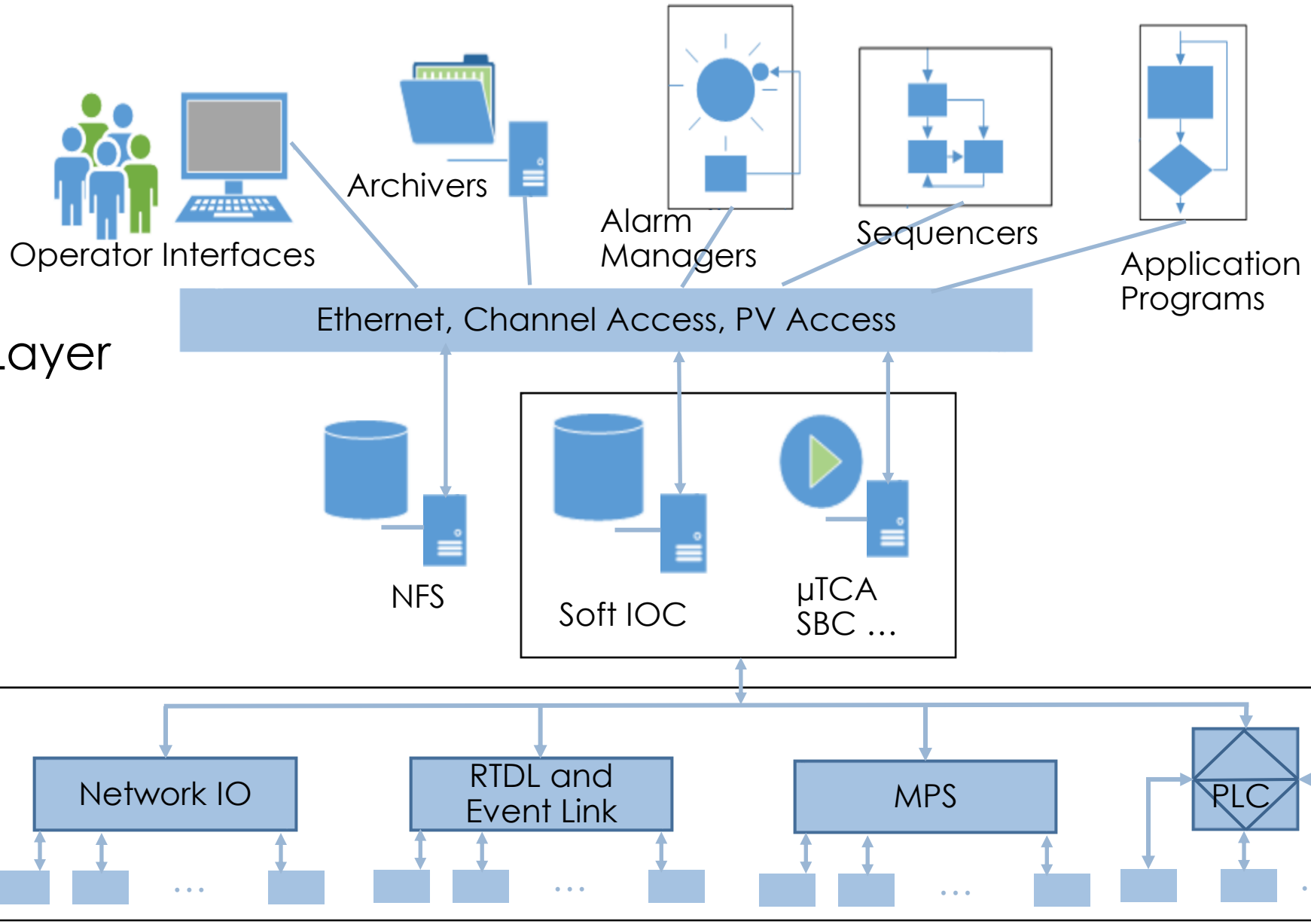
- Short pulses at **15 Hz** enables simultaneous **measurement across broad wavelength range** ($\Delta\lambda$ =13.2 Å at 20 m from source)
- Complementarity with FTS – **uses all available accelerator capability provided by PPU**
- **Flexibility** will be provided to **operate** both FTS and STS at the same time or separately if either is shutdown

ICS Scope

Integrated Control Systems (ICS) provides the integrated controls, data acquisition, computing infrastructure, and protection systems across all the STS technical areas



STS ICS Architecture: EPICS 7 based, 3-layer structure



Software Framework

- STS starts operating with the latest stable operating system and EPICS
  - Currently use Linux RHEL9
  - Currently use EPICS 7.0.7
  - Use the latest core modules (Asyn, StreamDevice, etc)
  - Use the latest CS-Studio (Phoebus)
- Software will be updated regularly during the project
- RHEL OS, EPICS base and support module versions will be frozen in advance of installation and commissioning

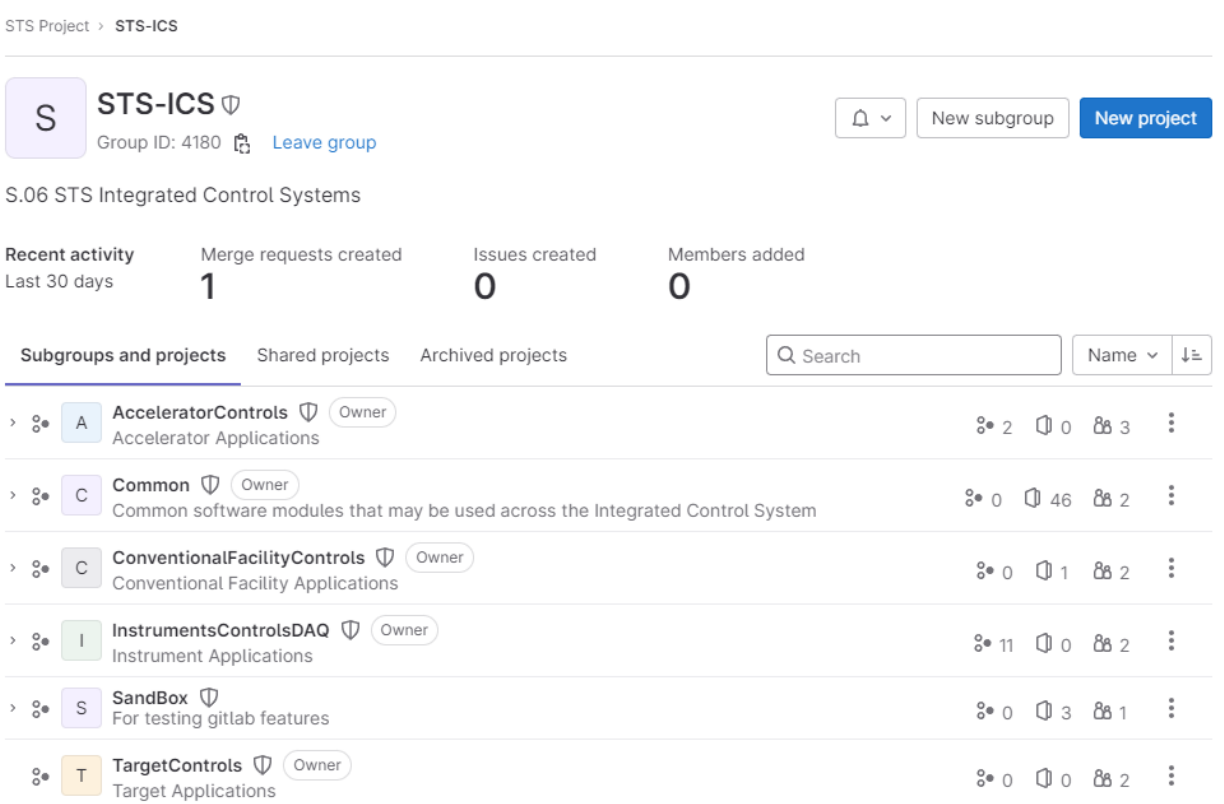
➔*PV Access will generally be supported alongside Channel Access!*

Software Development Environment

- Use of Git (branches, tags, vendor software)
- Git workflow diagrams
- Use of GitLab (groups, projects, permissions)

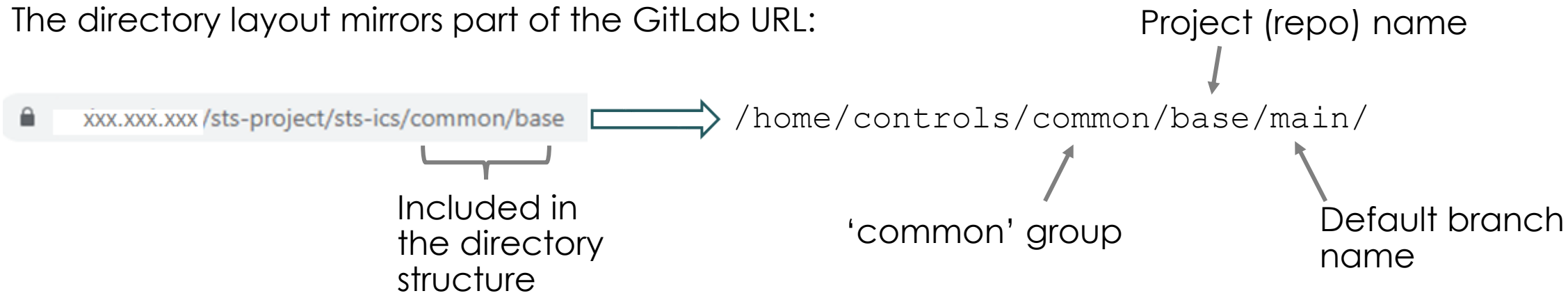
STS Project Layout (GitLab)

Groups for:
- Common
- Accelerator Controls
- CF Controls
- Instrument Controls & DAQ
- Target Controls



Software Deployment and Release

The directory layout mirrors part of the GitLab URL:



Use **ics-deploy** to build and deploy a project
Example (clone & build, or just rebuild):

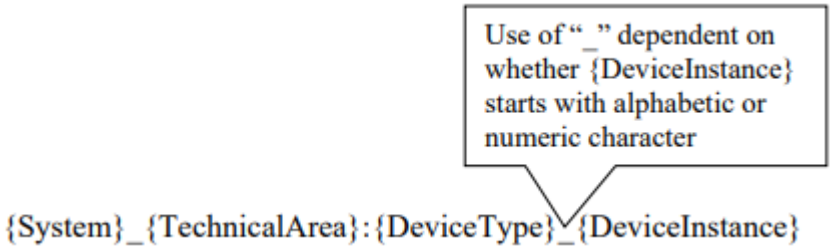
- `ics-deploy -b common/base main`
- `ics-deploy -b common/asyn main`

Use **sts-deploy** to release an application:

- The release goes into /home/controls/prod/ and is made read-only on successful build
- Can release IOC applications

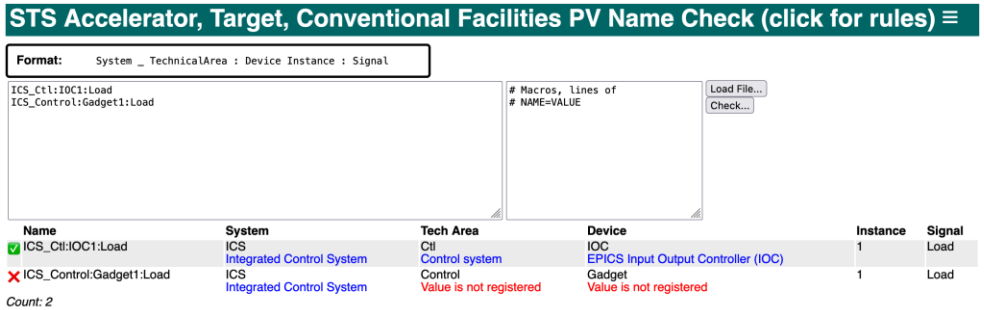
Device & Software Naming (Project Wide)

There is a device and PV naming standard for STS accelerator/target/CF.
Consistent with existing naming standard at SNS

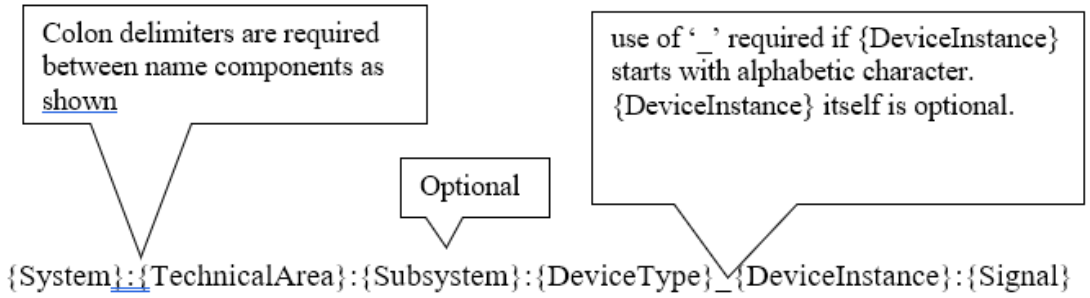


Examples:
RTST_Mag:QH01
RTST_Mag:PS_QH01
Tgt2_ACL1:Tnk21001

Name check web tool



There is a device and PV naming standard for STS instruments.
Consistent with existing naming practice at SNS



Examples:
ST01:Vac:CCG01
ST01:Chop:Mag:DCD01

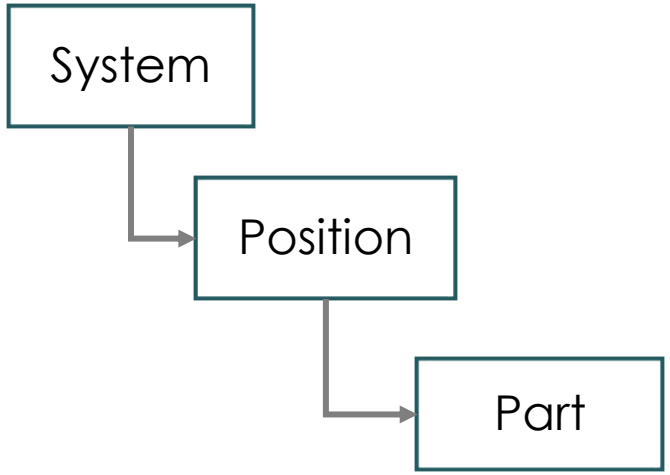
Names are managed in GitLab.
Names can be added via an approval process

Defined Device & Software Interfacing Standards

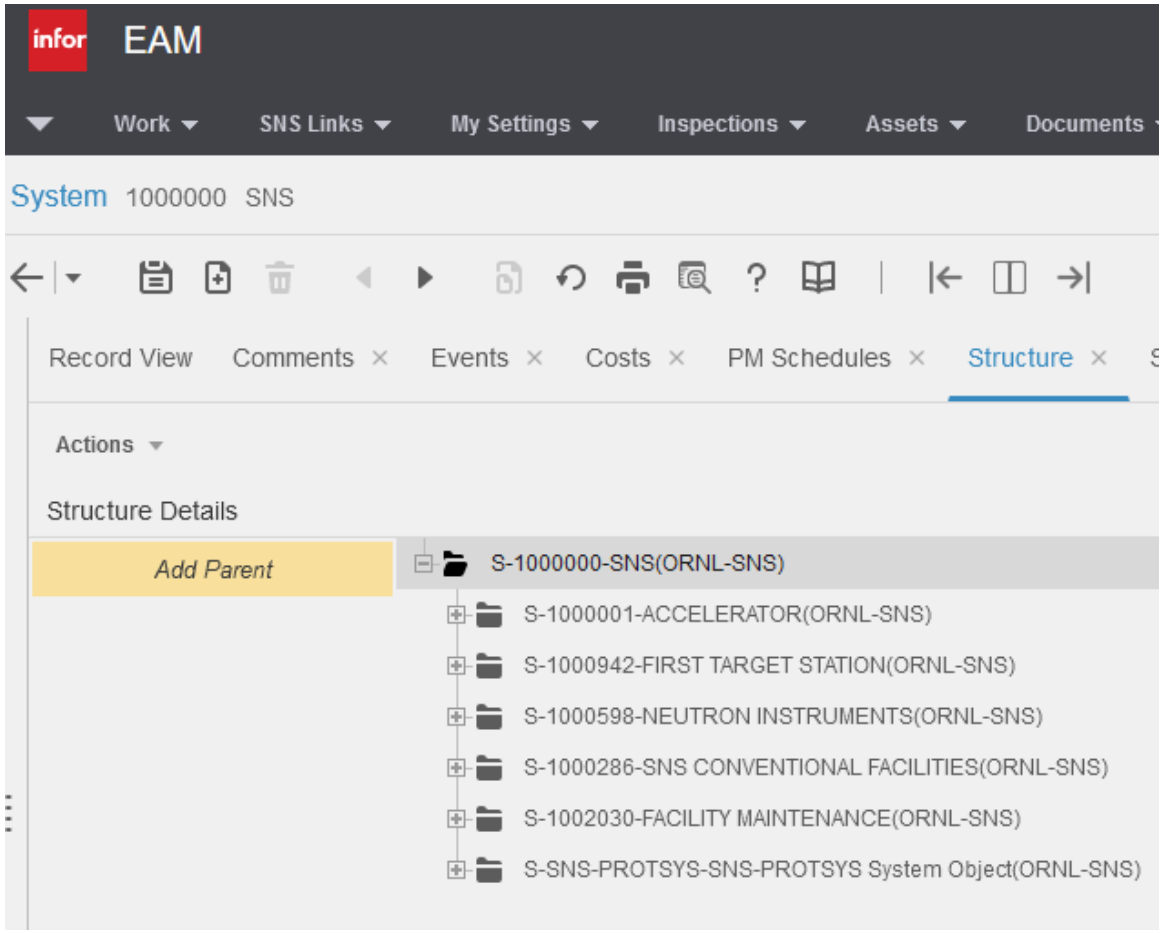
- Device Interfaces
  - Preferred connectors & Electrical Interfaces (e.g., ethernet, RS232 serial, analog and digital I/O,)
  - Preferred device/software communication protocols (e.g. EtherNet/IP, Modbus/TCP, ASCII)
- Recommended Devices
  - Preferred device list (e.g. Allen-Bradley PLC, Pfeiffer vacuum, Moxa device)
- Software Interfaces
  - Software application protocol preferences (e.g., EPICS channel access, TCP/IP)
  - Options for integrating LabVIEW and other types of systems

Asset Management

- Infor EAM (Enterprise Asset Management) will be used for equipment tracking and maintenance management



STS will have a similar structure to SNS



Conclusions

- Solid progress is being made on preliminary design to reach CD-2 milestone
- Integrated Control System design is based on existing SNS practice with improvements where necessary
- Implementation already in place to support multiple projects
- Confident that the control system infrastructure will work for the duration of the STS project and will satisfy STS requirements

Questions