

Integrated Control System Architecture for STS Project

Jay Yan, Steven Hartman, Kay Kasemir, Matthew Pearson

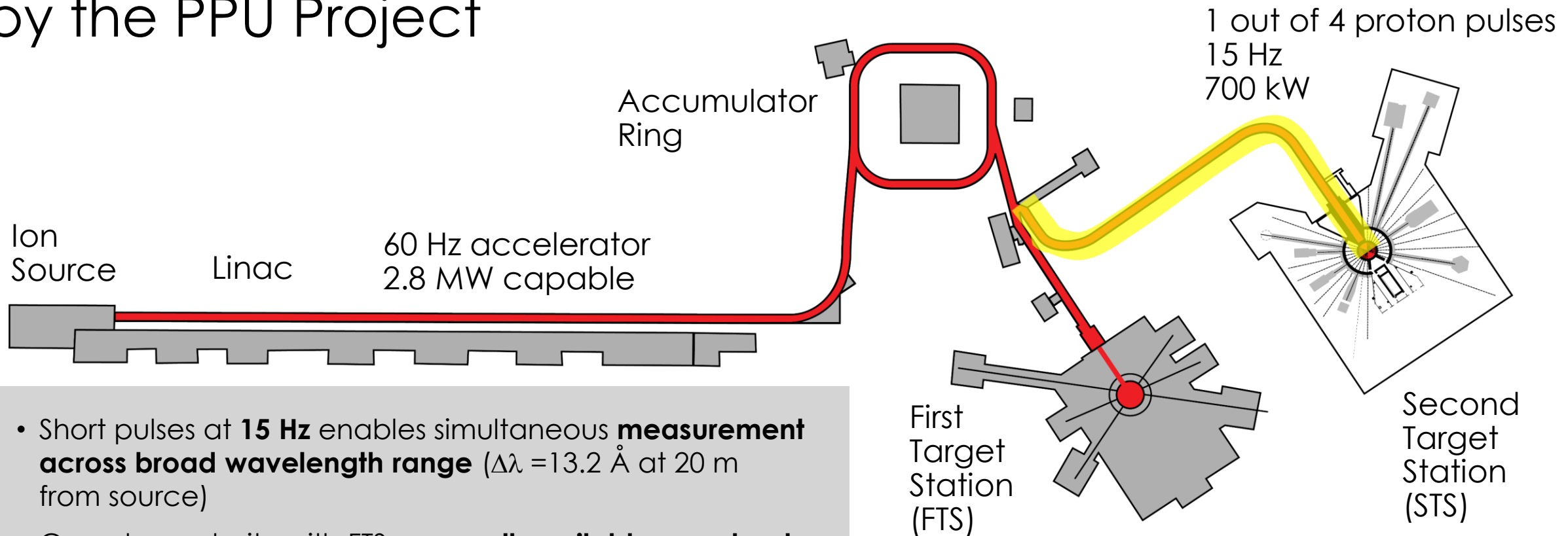
STS Integrated Control Systems

Oak Ridge National Laboratory

EPICS Collaboration Meeting, Cape Town, South Africa

Oct. 8, 2023

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

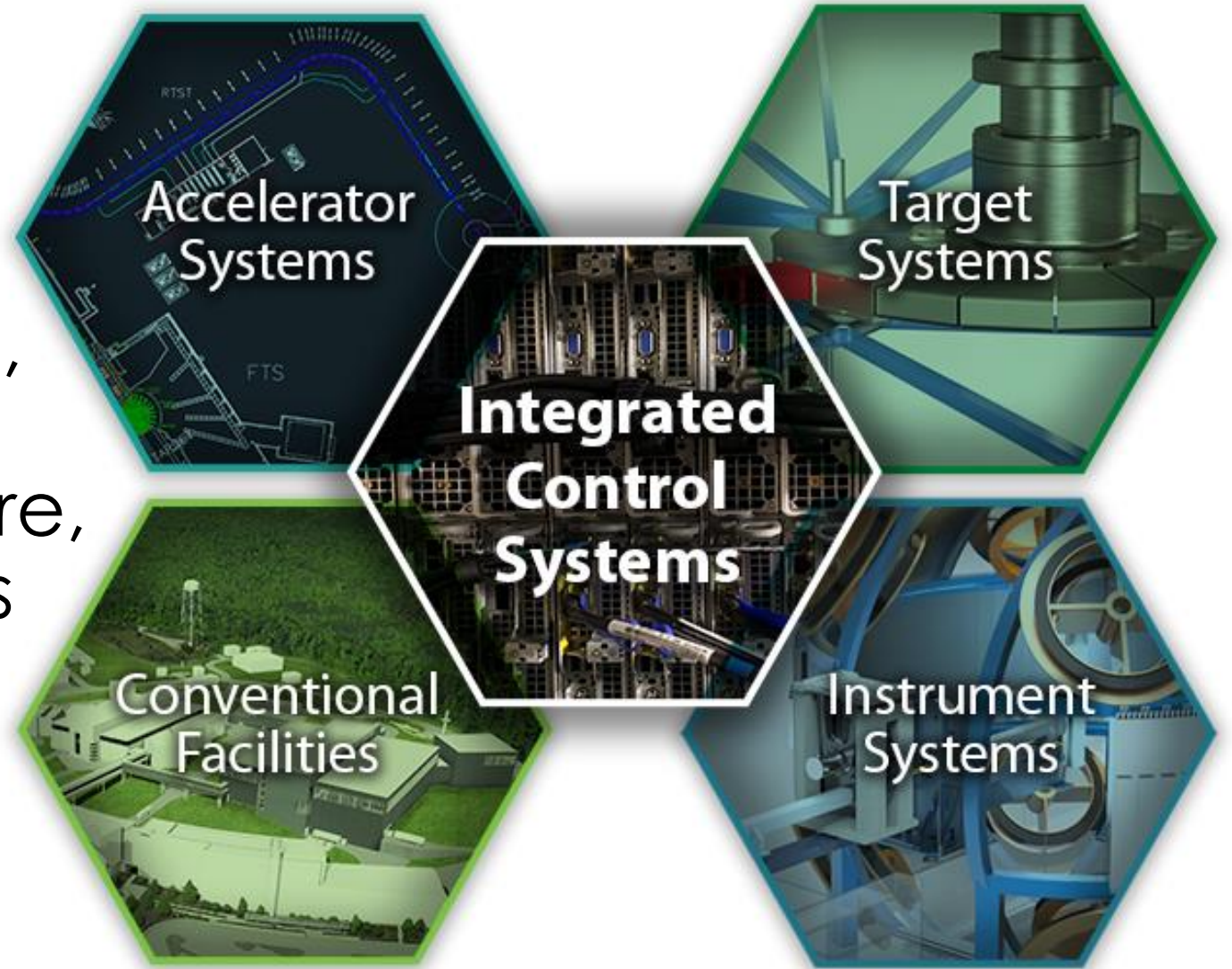


- Short pulses at **15 Hz** enables simultaneous **measurement across broad wavelength range** ($\Delta\lambda = 13.2 \text{ \AA}$ at 20 m from source)
- Complementarity with FTS – **uses all available accelerator capability provided by PPU**
- **Flexibility** will be provided to **operate** both FTS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

3 out of 4 proton pulses
45 pulses/second
2 MW

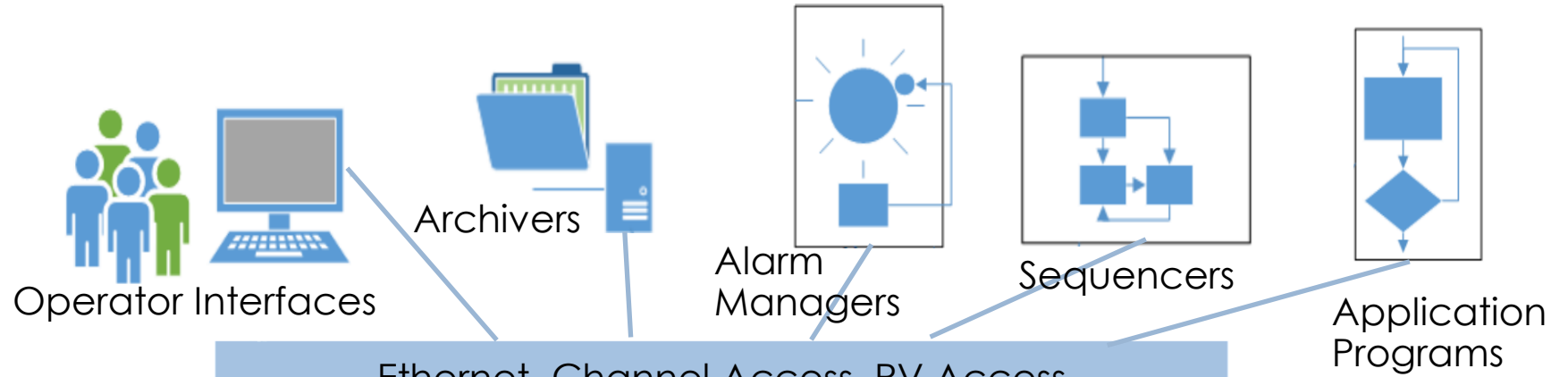
ICS Scope

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

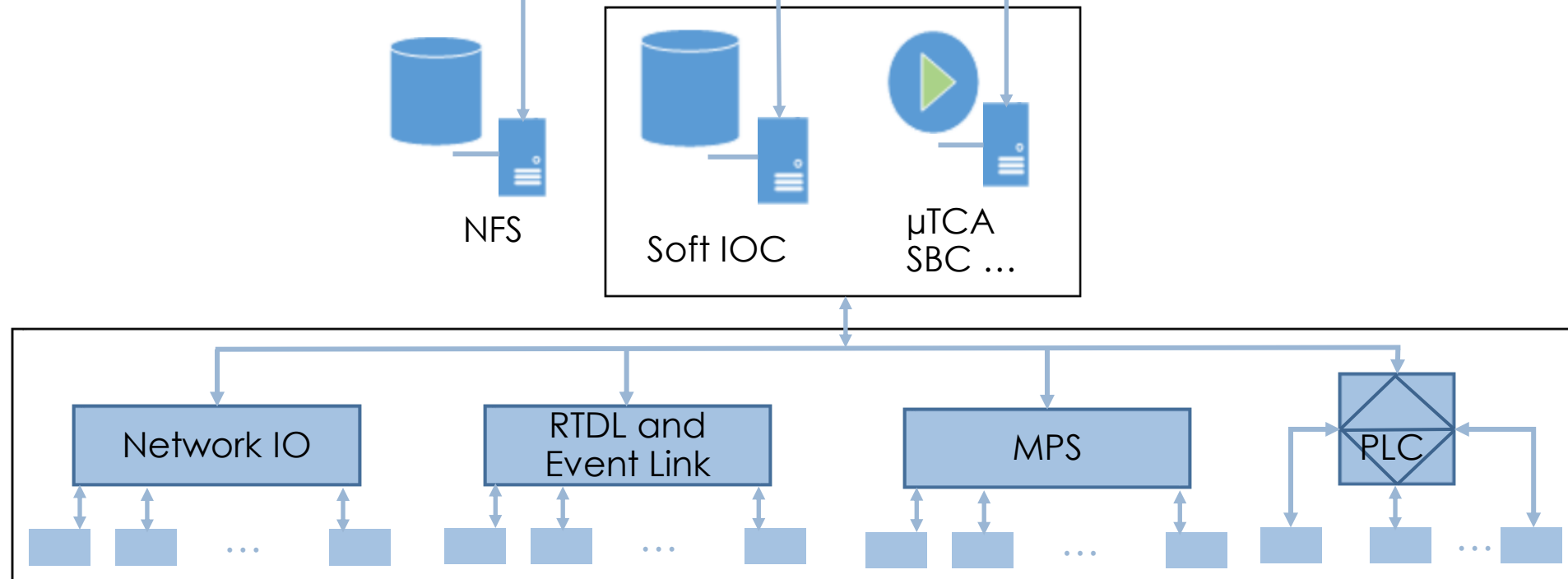
Back-end Layer



Communication Layer



Front-end Layer



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

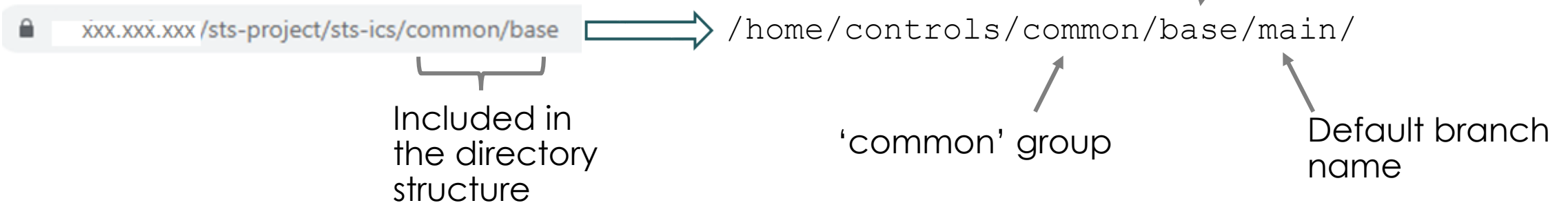
Target Controls

The screenshot displays the GitLab interface for the STS-ICS group. At the top, it shows the group name 'STS-ICS' with a shield icon, a group ID of 4180, and a 'Leave group' link. There are buttons for 'New subgroup' and 'New project'. Below this, the group description is 'S.06 STS Integrated Control Systems'. A summary section shows 'Recent activity Last 30 days' with 1 merge request created, 0 issues created, and 0 members added. A navigation bar includes 'Subgroups and projects', 'Shared projects', and 'Archived projects', along with a search bar and a 'Name' dropdown. The main content area lists several subgroups and projects:

Subgroup/Project Name	Members	Issues	Projects
AcceleratorControls (Accelerator Applications)	2	0	3
Common (Common software modules that may be used across the Integrated Control System)	0	46	2
ConventionalFacilityControls (Conventional Facility Applications)	0	1	2
InstrumentsControlsDAQ (Instrument Applications)	11	0	2
SandBox (For testing gitlab features)	0	3	1
TargetControls (Target Applications)	0	0	2

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

Use of “_” dependent on whether {DeviceInstance} starts with alphabetic or numeric character

Examples:
RTST_Mag:QH01
RTST_Mag:PS_QH01
Tgt2_ACL1:Tnk21001

{System}_{TechnicalArea}:{DeviceType}_{DeviceInstance}

Name check web tool

STS Accelerator, Target, Conventional Facilities PV Name Check (click for rules) ≡

Format: System _ TechnicalArea : Device Instance : Signal

ICS_Ctl:IOC1:Load
ICS_Control:Gadget1:Load

Macros, lines of # NAME=VALUE

Load File...
Check...

Name	System	Tech Area	Device	Instance	Signal
✓ ICS_Ctl:IOC1:Load	ICS Integrated Control System	Ctl Control system	IOC EPICS Input Output Controller (IOC)	1	Load
✗ ICS_Control:Gadget1:Load	ICS Integrated Control System	Control Value is not registered	Gadget Value is not registered	1	Load

Count: 2

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

Colon delimiters are required between name components as shown

Optional

use of ‘_’ required if {DeviceInstance} starts with alphabetic character. {DeviceInstance} itself is optional.

{System}_{TechnicalArea}:{Subsystem}:{DeviceType}_{DeviceInstance}:{Signal}

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

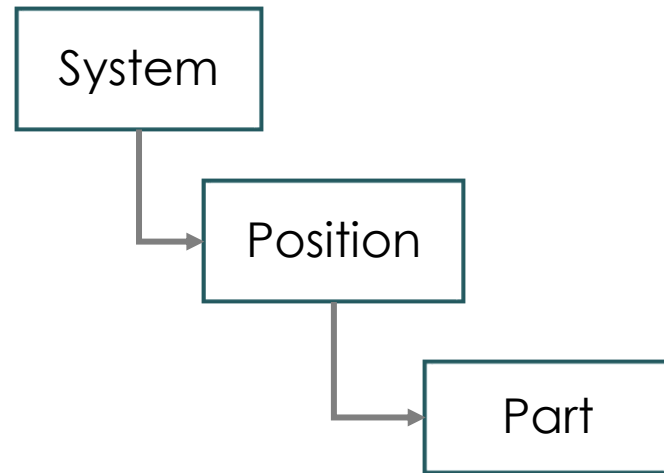
Names are managed in GitLab.
Names can be added via an approval process

Defined Device & Software Interfacing Standards

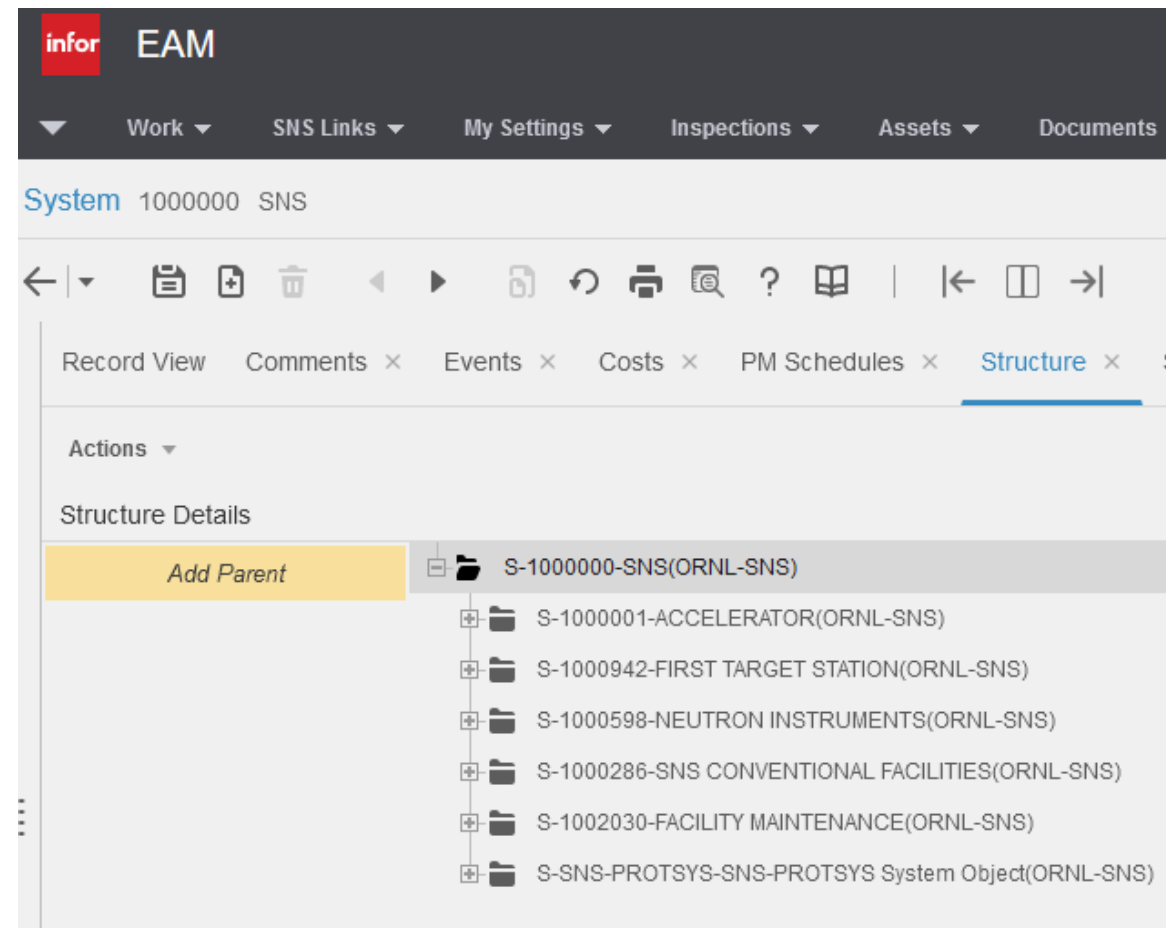
- Device Interfaces
 - Preferred connectors & Electrical Interfaces (e.g., ethernet, RS232 serial, analog and digital I/O,)
 - Preferred device/software communication protocols (e.g. EtherNet/IP, Modbus/TCP, ASCII)
- Recommended Devices
 - Preferred device list (e.g. Allen-Bradley PLC, Pfeiffer vacuum, 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

