EUROPEAN
SPALLATION
SOURCE

Status of the European Spallation Source Control System

Quick overview

PRESENTED BY TIMO KORHONEN, REPRESENTING THE INTEGRATED CONTROL SYSTEM DIVISION

2023-10-13

Agenda

1 Introduction

2 Standards and Platforms

3 Development philosophy, tools and goals

4 Keeping in line

5 Lessons learned – so far

6 Future work & road to project completion

7 Acknowledgements

The European Spallation Source ERIC

An accelerator-based neutron source in Lund, southern Sweden and

Data Management Center (DMSC) in Copenhagen, Denmark

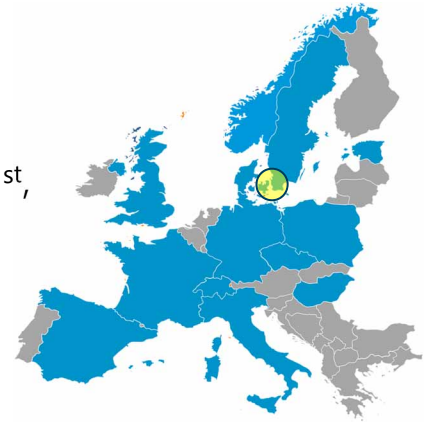
- Material and life sciences research

A collaboration of 13 European nations

- An ERIC (European Research Infrastructure Consortium) since October 1st, 2015
- Superconducting linac, rotating tungsten target
- 15 neutron instruments in construction,

Key parameters to be reached by start of user program

- 2 MW beam power capability, 2 GeV proton energy, 14 Hz repetition rate, 2.86 ms pulse@62.5 mA beam current
- Beam power to be increased as budget allows (5 MW)
- More neutron instruments/beamlines to be added



Construction progress

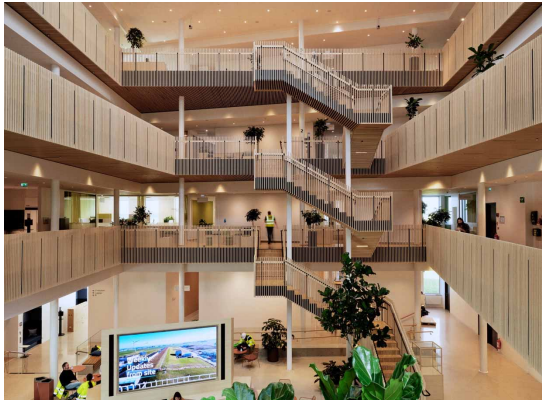
At the time of ICALEPCS 2015, Melbourne
Green field…

About 2019, just before covid pandemic
Starting to take shape

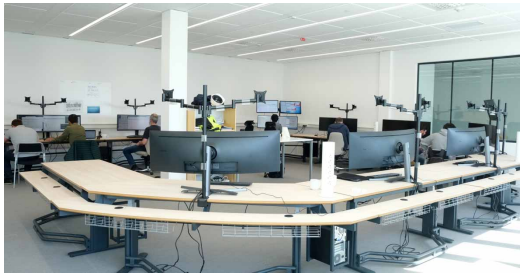
ESS site in 2022…

Coming out of the pandemic

- All buildings ready and handed over.
- Equipment installation continues
  - Accelerator installation continues
  - Target and neutron beamlines in very active construction



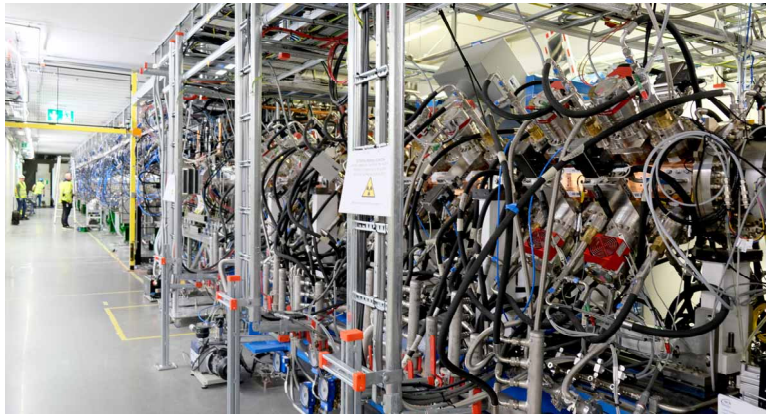
- We moved in the new offices



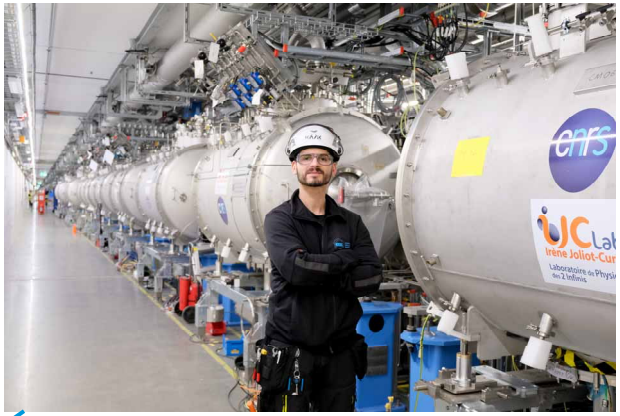
- The main control room, to be equipped

How far have we come?

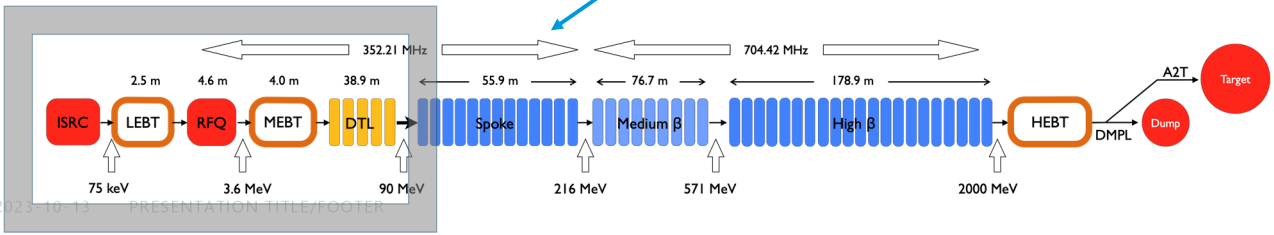
Sub-headline to strengthen the headline above



Normal Conducting Linac. Commissioned in 2023 (minus one DTL tank)



Installation of superconducting cavities (cryomodules) ongoing.



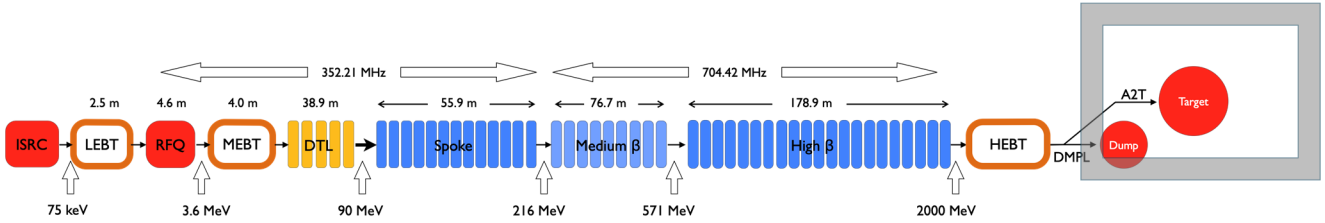
How far have we come?

Overall progress

Servers & IT

Target station “monolith”

Neutron instrument hall



2023-10-13 PRESENTATION TITLE/FOOTER

The Control System

Scope and references to presentations

Computing infrastructure: Networks, server infrastructure, OS

- CentOS 7, now migrating to Ubuntu

Hardware

- **THMBCMO21 Development of Standard MicroTCA Deployment at ESS**

Timing system

- **THMBCMO24 Time Synchronization and Timestamping for the ESS Neutron Instruments**
- **THPDP051 LLRF and Timing System integration at ESS**

Subsystem integration (EPICS IOCs, PLC-based systems)

- **THPDP053 Test Automation for Control Systems at the European Spallation Source**
- **THPDP052 Characterizing Motion Control Systems to Enable Accurate Continuous and Event-Based Scans**
- **TUPDP079 Commissioning of PLC Based Control System for MBL Cryomodules at ESS SRF Linac**
- **TUPDP080 Automated Procedure for Conditioning of Normal Conducting Accelerator Cavities**
- **THMBCMO11 Full Stack PLC to EPICS Integration at ESS**

The Control System

Scope and references to presentations

Control room tools:

- CS-Studio Phoebus, Logbook (olog), Save&Restore,

Common services:

- Archiving: Archiver Appliance
- Channel Finder (Record dictionary),

Machine protection

- **TU2BCO06 Verification and Validation of the ESS Machine Protection System-of-Systems (MP-SoS)**
- **TUPDP081 The ESS Fast Beam Interlock System - Design, Deployment and Commissioning of the Normal Conducting Linac**

Personnel protection

- **TUMBCMO37 Personnel Safety Systems for ESS Beam on Dump and Beam on Target Operations**
- **TUPDP078 Management of the Change Control Process for Protection Systems at ESS**

The Control System

Getting started…

Challenges:

- Many participating institutes and suppliers
- Growing organisation, with developers coming in from different backgrounds

How to:

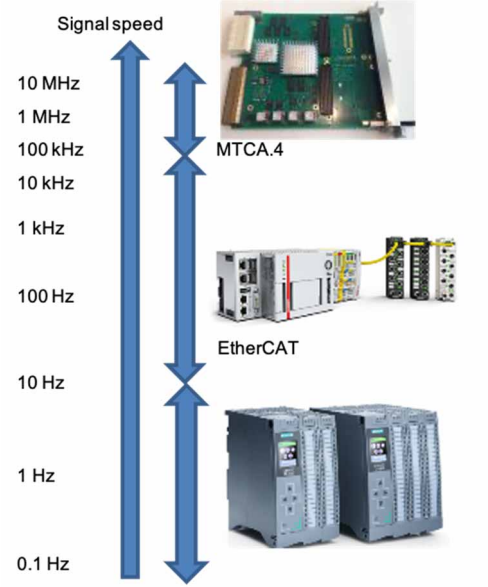
- Keep overall consistency and achieve a maintainable status
- Fulfil the expectations (internal & external) to be state-of-the-art
- Be prepared for continuous evolution

Set standards and build support structures

- Was a challenge in a new organisation, high fluctuation of personnel

Standards – hardware platforms

Set early to keep consistency



MTCA.4 for high-bandwidth applications

- **THMBCMO21 Development of Standard MicroTCA Deployment at ESS**
- Will have ~300 MTCA systems in the final configuration
- A pity is that there are “camps” with limit interoperability of components
  - Could we as the community do something? Open source developments to the rescue? Push manufacturers?

PLCs and EtherCAT with and without PLC fill their respective ranges well

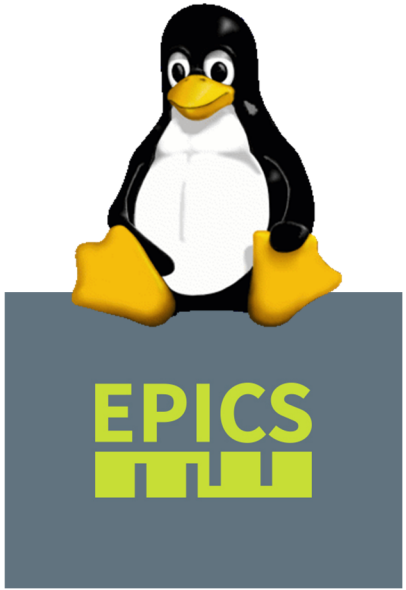
- **THMBCMO11 Full Stack PLC to EPICS Integration at ESS**
- **THPDP052 Characterizing Motion Control Systems to Enable Accurate Continuous and Event-Based Scans**

In general, adoption has been good.

- A couple of minor exceptions that ~~can~~ have to be sorted out.
- No hardware zoo, as we feared

Standards – software side

Selection of the fundamental components



Linux CentOS 7 -> Ubuntu

EPICS 7 + e3 environment

CS-Studio -> Phoebus

Development and deployment systems

GitLab

ANSIBLE AWX

cs|studio

Development

Tools and goals #1

In 2015 ICALEPCS 2015 contribution we wrote:

"Our goal is to build the system based on the EPICS Version 4. ***All our low-level controllers as well as software applications use the version 4 structured data and normative types at the application level and communicate using pvAccess.***"

To push towards that goal, we wanted to be active in the EPICS community

- Funding developments and developing internal know-how
- Get involved in the "core" EPICS development as well as client tools

In the meantime, EPICS V4 became EPICS 7.

For all practical purposes we can say that we have reached the goal this spring.

- As presented in the EPICS workshop on Saturday. Not 100% yet, but close.

Development

Tools and goals #2

Like all projects, we needed to have a development environment

How to get that up in a short time in a rapidly evolving environment?

- Try to look at existing concepts.

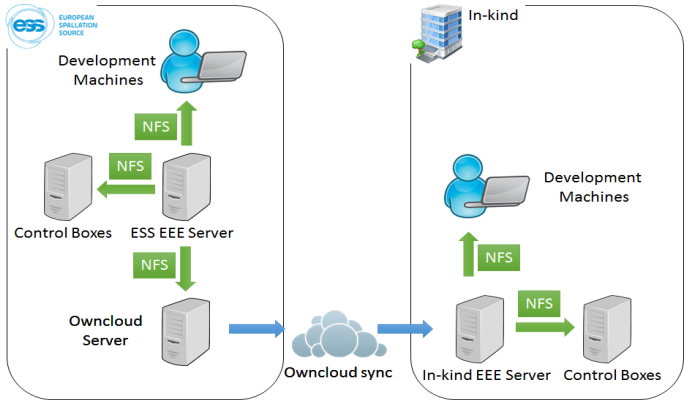
Some additional goals:

- Threshold to update should be as low as possible
  - Keep up with new developments
  - Enable quality control as far as possible
  - Frequent small updates instead of a “big bang”
  - System-wide features (e.g., access control, monitoring, channel finder) have to be consistently set up to be useful.

Started with a concept from PSI.

Development - early

Tools and goals



As presented in ICALEPCS 2015

Early idea of "cloud-based" development

Worked – almost.

- Deployment processes immature
- Different infrastructures and expectations at partner labs
- Lack of understanding of the underlying tools
  - Copied from PSI

Rework was necessary

- Needed to diverge, and study.

But the fundamental idea was not abandoned:

- Small but frequent updates

Development

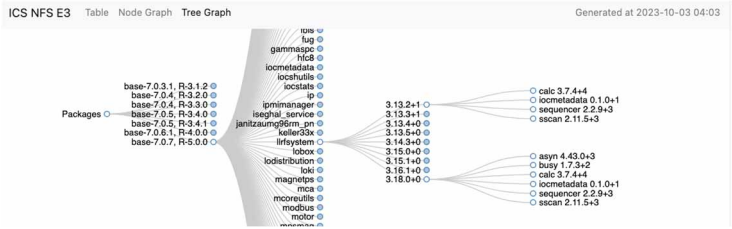
Tools and goals

The third generation of our EPICS Environment e3

- **MO4BCO05 Apples to Oranges: A Comparison of EPICS Build and Deployment Systems**

- Inflicted pain to an internal group of developers, but achieved:
  - IOC instances become very simple
  - Modules (defining functionality) are centrally managed and there is a gatekeeper
  - Separation of concerns: less load for integrators who do not have to worry about writing makefiles and collecting together software pieces
  - Modules that provide system integration are centrally managed and updated
- Somewhat unorthodox way to build EPICS IOCs, but at least if you ask me, this works extremely well
  - But you should ask users what they think
- Significantly lowers the barrier to keep IOCs up to date with developments in EPICS

EPICS IOC	E3 IOC
Run makeBaseApp	
Define Base, Modules in RELEASE	
Add database and protocol files	Add database and protocol files
Update Makefile	
Build	
Edit st.cmd	Write st.cmd[1]
Run	Run

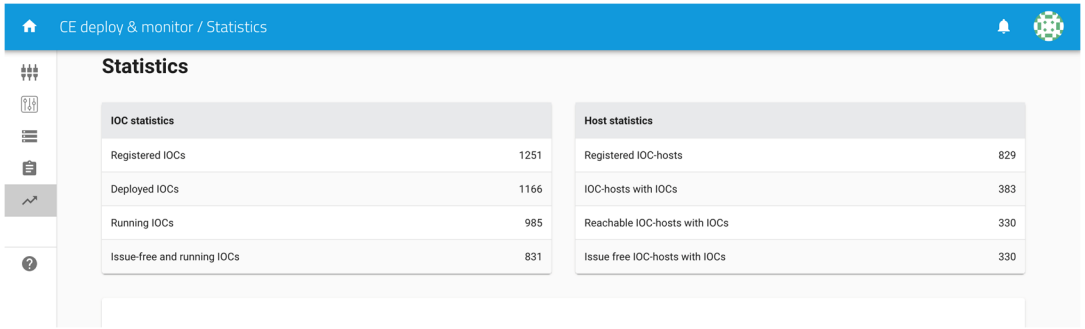


Deployment

How to maintain an overview of thousands of IOCs?

Development takes us half the way. System have to be deployed to production.

- This can be done in countless ways… but we want also to be able to
  - Have an overview of what is out there: what, when, by whom
  - See what the IOCs are doing (system logs, putlog,…) and what they provide (EPICS PVs)
  - Be able to stop/start/restart them
- This is what our deployment system provides, provided:
  - a common “infrastructure” module is included in all IOCs
    - Provides recsync (Channel Finder), security settings, autosave,…
  - IOCs are (reasonably) up to date. Only then the picture is complete.

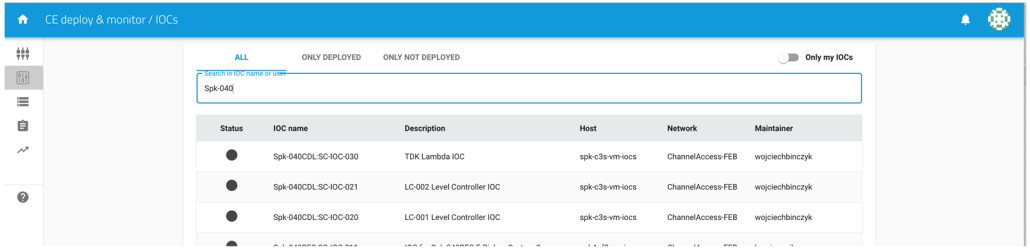


985 of 1166 deployed IOCs running, on
383 hosts (mostly virtual)

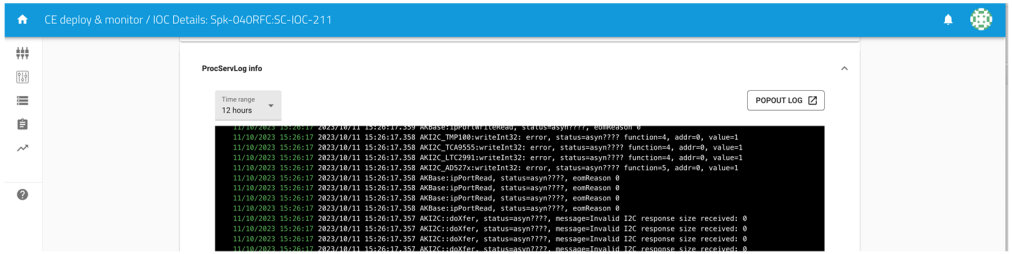
Some IOCs have “issues” (missing or incomplete configuration, etc)
As well as some hosts.

Deployment system

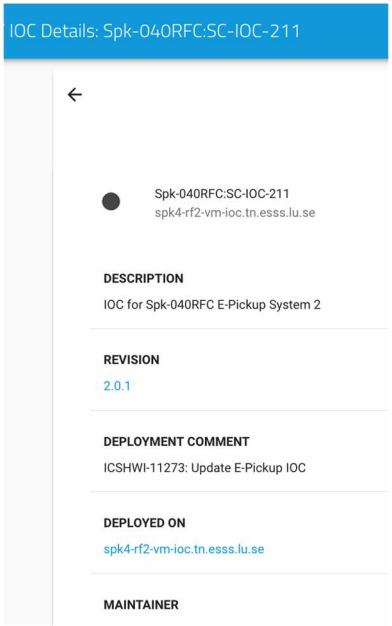
Features showcase (a selection)



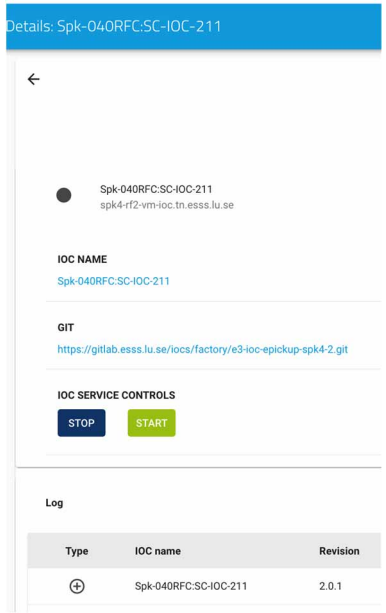
List and search for IOCs



Direct access to system logs



Deployment info
Link to revision



Management
Link to Git repo

Lessons learned – so far

Mostly my personal reflections

Setting overarching goals has paid off
- We did not get into the mess we feared.
- People have played well with the rules – when they were clear

Communication
- Do not expect that you will be immediately understood
- Repeat and cross-check common understanding

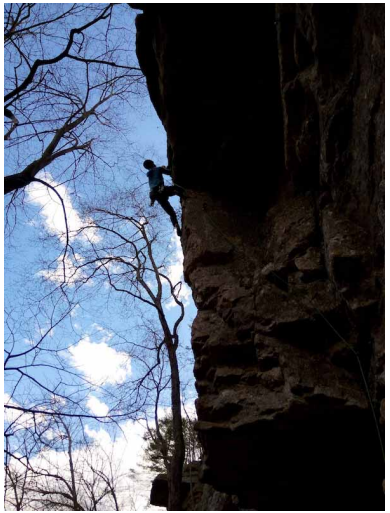
Listen to your stakeholders.
- But sometimes, what the stakeholders want is not the same as they need.
- Again, spend sufficient time to understand the issues

Be prepared to change course when needed
- Not too quick though. When you are convinced that there is a better alternative

Future work and conclusions

The road to user operation

- Basics of the control system are in place and working
  - We could commission a substantial part of our machine with relatively few major issues
  - EPICS 7 and pvAccess work well
  - Data volumes are an issue – need more intelligent ways to handle appropriately
- Integration on larger scale needs to be ~~improved~~ implemented
  - Adding automated sequences to handle routine sequences
  - Improves reliability and speeds up machine setup
  - Critical for the next stages – complexity has grown too large to handle “manually”.
- Lots of awesome work has been done by my colleagues in-house and in the community
  - Has felt like mountain climbing – but the top is getting closer
- Future will show, but I think we have a good basis for the years to come.



Thank you for your attention!