
ess
EUROPEAN
SPALLATION
SOURCE

So much of it!

Operations data at the European Spallation Source

DARREN SPRUCE, EUROPEAN SPALLATION SOURCE
IBERGRID 2026

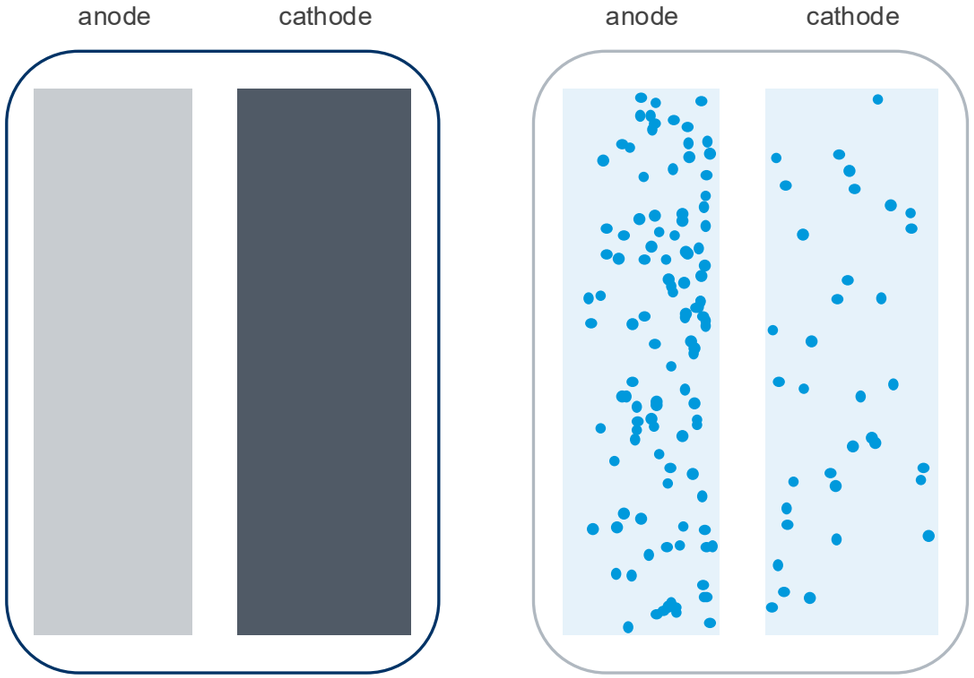
Why neutrons?

They see what X-rays cannot: light atoms inside heavy materials

Pass through metal
No charge, so they penetrate steel and aluminium casings

See light elements
Hydrogen and lithium show up strongly; X-rays barely see them

Non-destructive
Study a real device while it is working



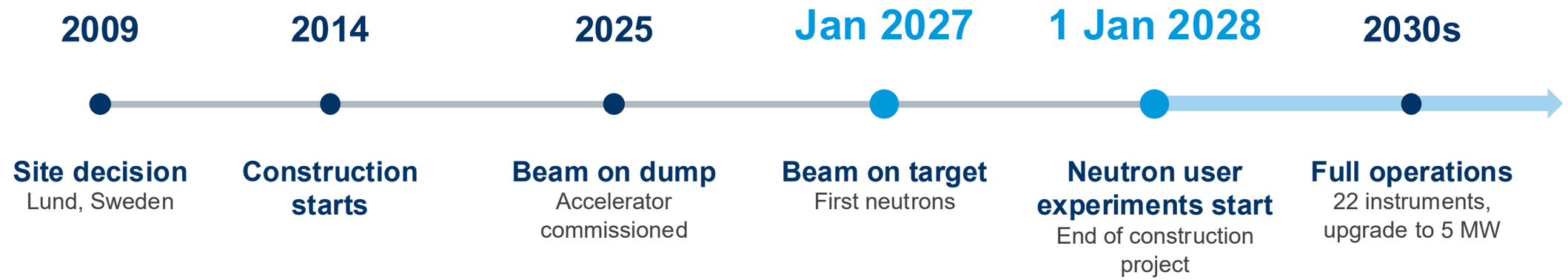
X-rays
casing and heavy metals

Neutrons
where the lithium is

Illustration: lithium moving between electrodes inside a sealed cell, seen while it charges

The ESS road to science

From choosing a site to neutron science, and onward



The control system has been recording since commissioning began.

How it works in a nutshell

1 Protons are generated in the ion source

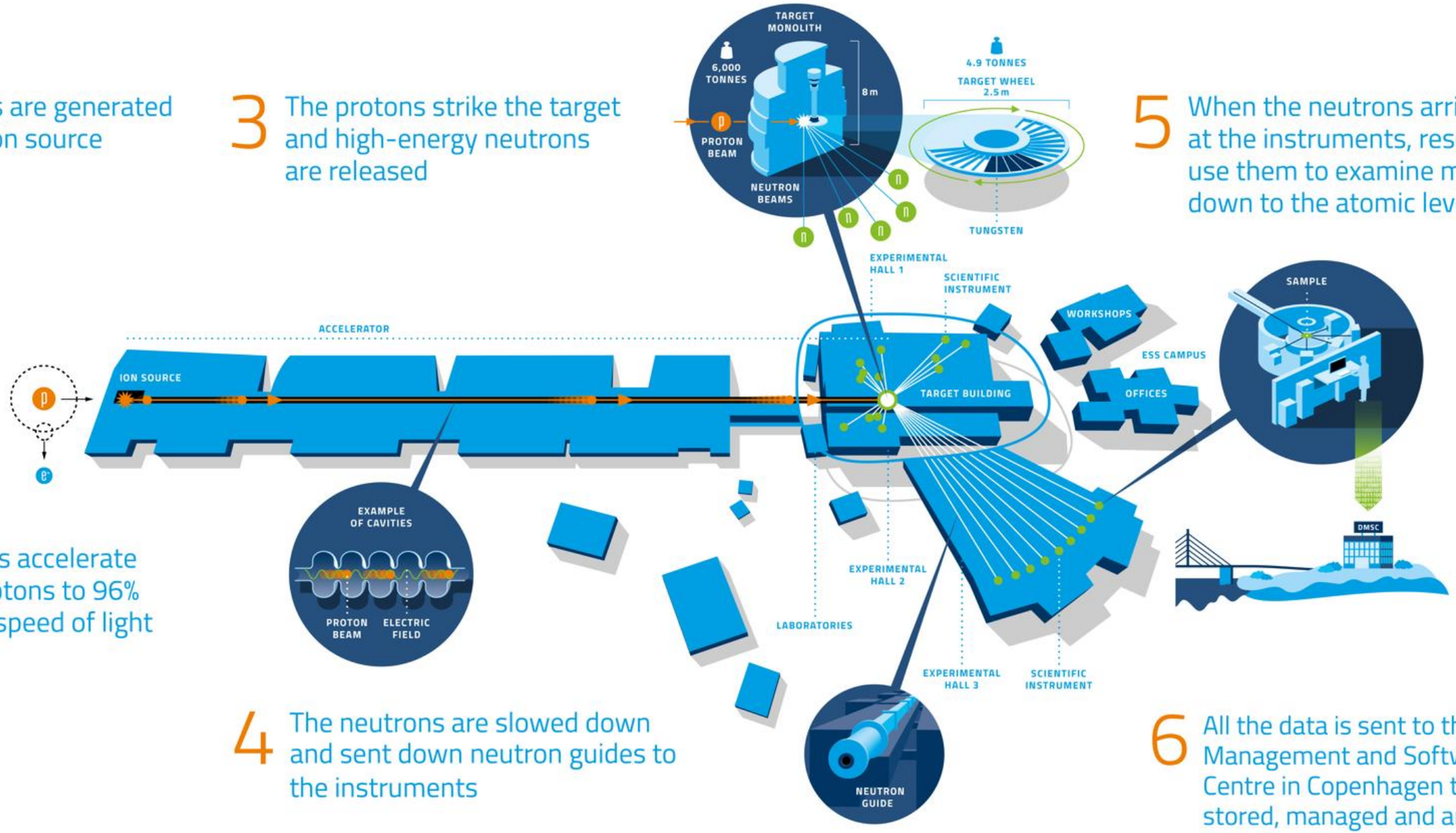
2 Cavities accelerate the protons to 96% of the speed of light

3 The protons strike the target and high-energy neutrons are released

4 The neutrons are slowed down and sent down neutron guides to the instruments

5 When the neutrons arrive at the instruments, researchers use them to examine matter down to the atomic level

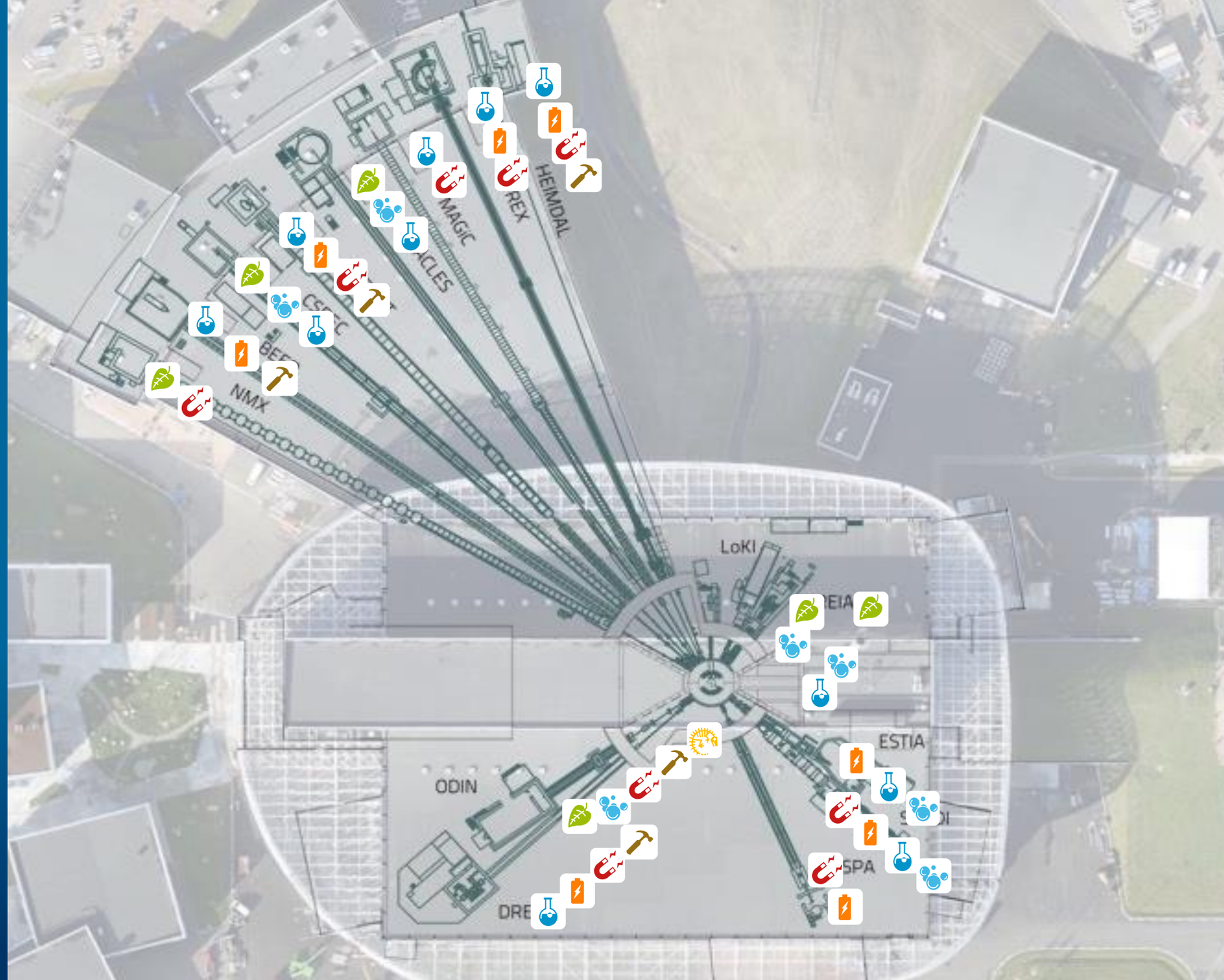
6 All the data is sent to the Data Management and Software Centre in Copenhagen to be stored, managed and analysed with the researchers



15 instruments under construction

Each instrument designed to be world-leading at 2MW

- Life sciences
- Soft condensed matter
- Chemistry of materials
- Energy research
- Magnetism & superconductivity
- Engineering & Geo-sciences
- Archeology & Heritage conservation



13 European countries joining forces and sharing a common ambition

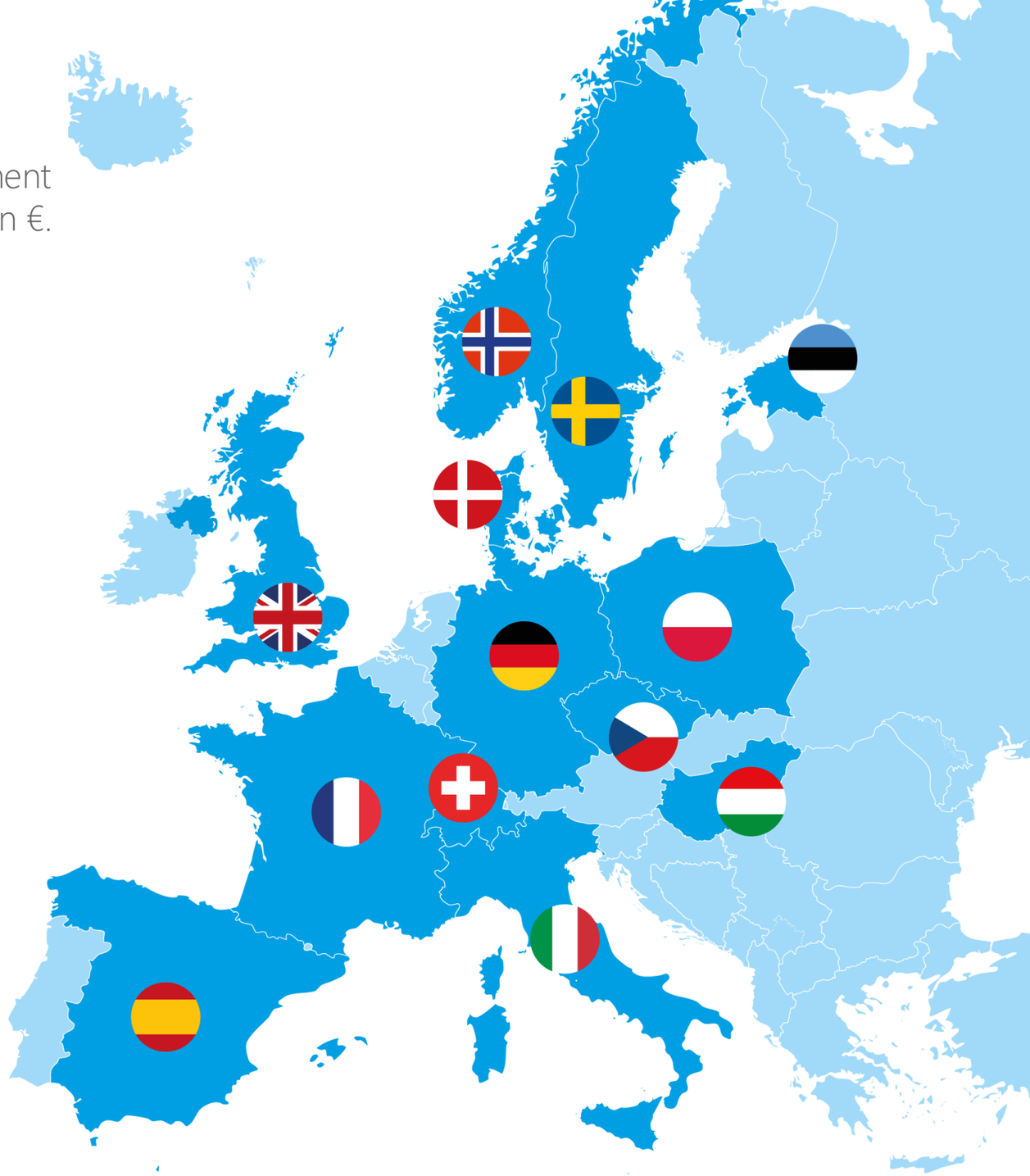
A 3 Billion € project, with an estimated annual operations and development cost of close to 200 million €.

Host States:

Sweden, **Denmark**.

Member States:

Czech Republic, Denmark, Estonia, France, Germany, Hungary, Italy, Norway, Poland, Spain, Sweden, Switzerland, United Kingdom



A unique international project >100 in-kind partners

Aarhus University
Atomki - Institute for Nuclear Research
Bergen University
CEA Saclay, Paris
Centre for Energy Research, Budapest
Centre for Nuclear Research, Poland, (NCBJ)
CNR, Rome
CNRS Orsay, Paris
Cockcroft Institute, Daresbury
Copenhagen University
Elettra – Sincrotrone Trieste
ESS Bilbao
Forschungszentrum Jülich
Helmholtz-Zentrum Geesthacht
Huddersfield University
IFJ PAN, Krakow
INFN, Catania
INFN, Legnaro
INFN, Milan
Institute for Energy Research (IFE)
Laboratoire Léon Brilouin (CEA/CNRS/LLB)
Lund University
Nuclear Physics Institute of the ASCR
Oslo University
Paul Scherrer Institute (PSI)
Polish Electronic Group (PEG)
Roskilde University
Rutherford Appleton Laboratory, Oxford (ISIS)
Tallinn Technical University
Technical University of Denmark
Technical University Munich
Science and Technology Facilities Council
UKAEA Culham
University of Tartu
Uppsala University
WIGNER Research Centre for Physics
Wroclaw University of Technology
Warsaw University of Technology
Zurich University of Applied Sciences (ZHAW)



64 nationalities

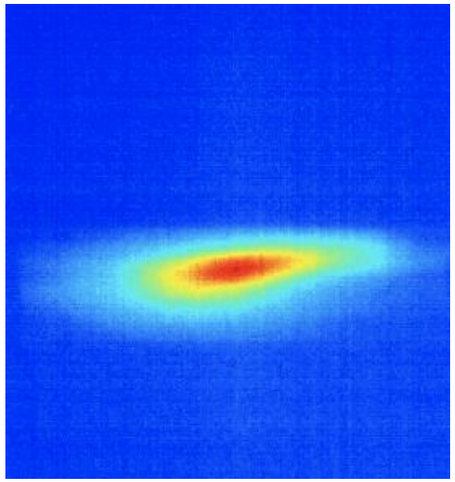
ESS' strongest asset as a scientific and technological powerhouse will be its workforce

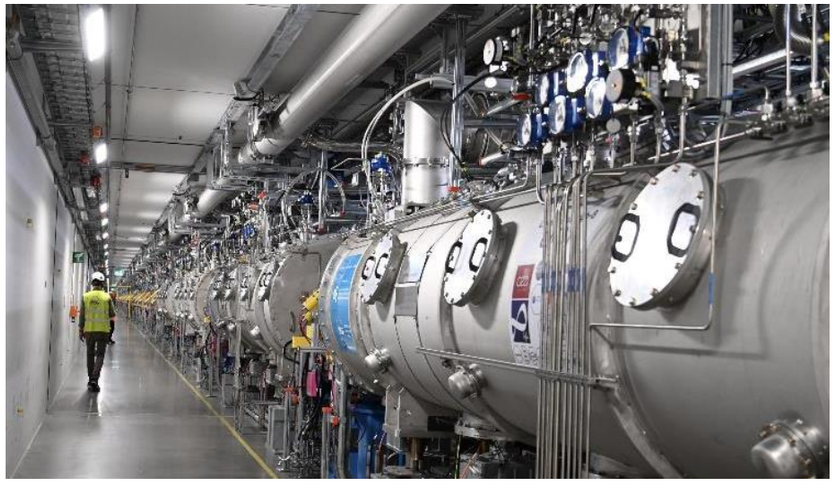


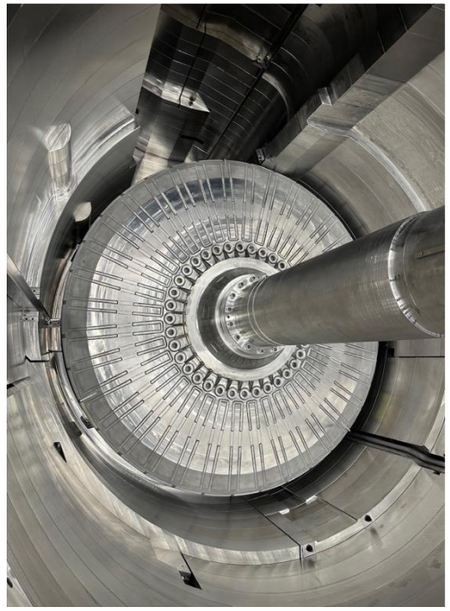
Highlights 2025

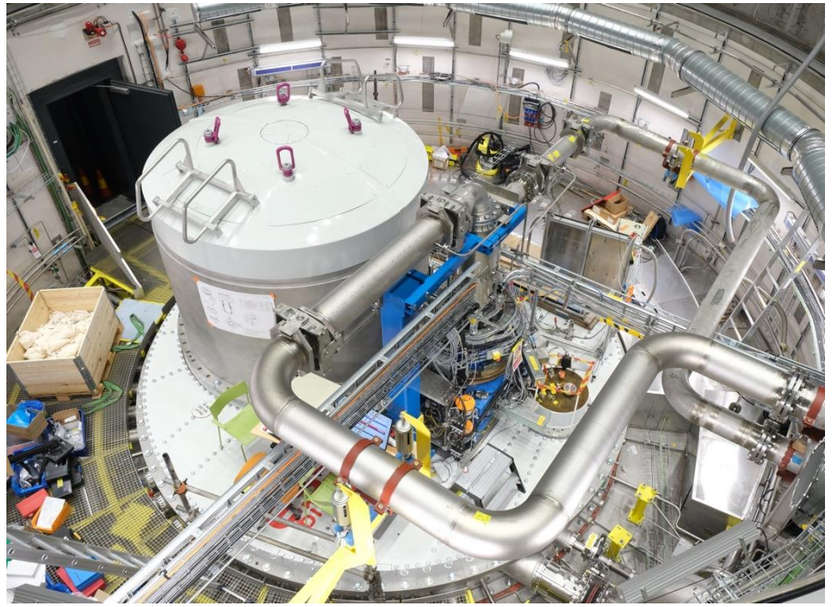
Commissioning of the accelerator (Beam on Dump)

All major components of the target system installed, in preparations for **Beam on Target**









Highlights 2025

Four instruments ready for neutrons

Outreach to the User community, and readying support labs

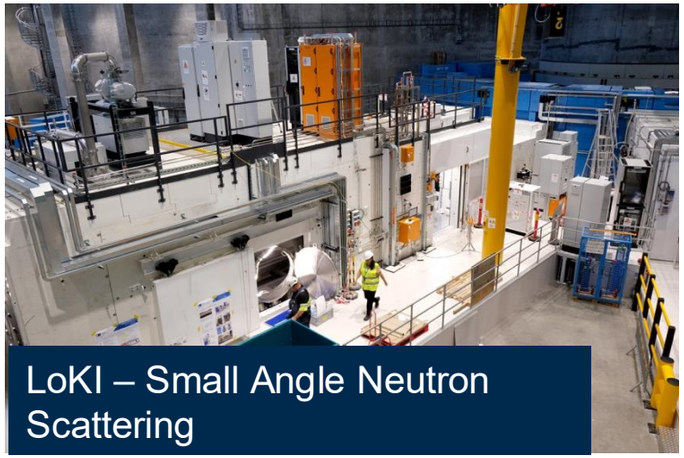
Major steps towards readiness for operations across the whole organisation



TBL - Test Beamline



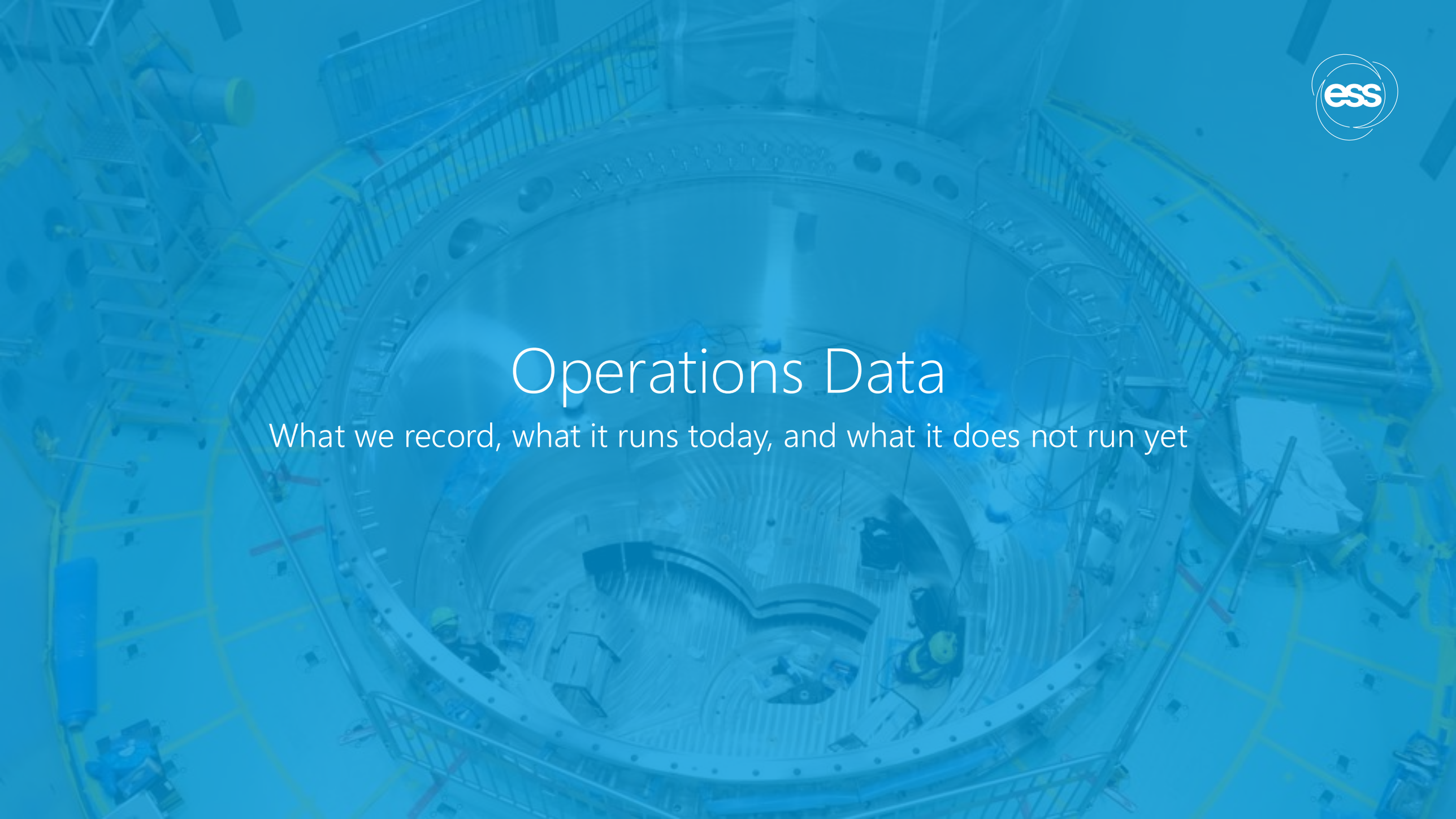
BIFROST – Extreme environment spectroscopy



LoKI – Small Angle Neutron Scattering



ODIN – Multi-purpose Imaging



Operations Data

What we record, what it runs today, and what it does not run yet

1.6M signals archived, 11.5M measurable

Among the largest operations archives at any accelerator facility

11.5M

control points the control system can expose and measure

1.6M

signals actually archived today; the rest could be added, but probably never will be

3 PB

long-term storage capacity, fed by tiered NVMe and disk

Temperatures, vacuum pressures, magnet currents, RF cavity phases, beam positions, interlock states — what is archived is kept for the operating lifetime of the facility under formal stewardship directives.

Layer Architecture of the ESS Control Systems

EUROPEAN SPALLATION SOURCE

Human Users

engineers, operators, scientists, technicians, ...

display ESS facility data, task execution

Applications

GUIs for data analysis, processing and archiving services, system configuration, ...

ESS facility data exchange

Facility Integration

runtime & historical data, complex control functions, configuration databases

system specific data exchange

Local System Control

process and device control, high dependability, executes local control functions

local control, run-time data

Signal Conditioning (a/d) and Electronics

Analogue and digital signals

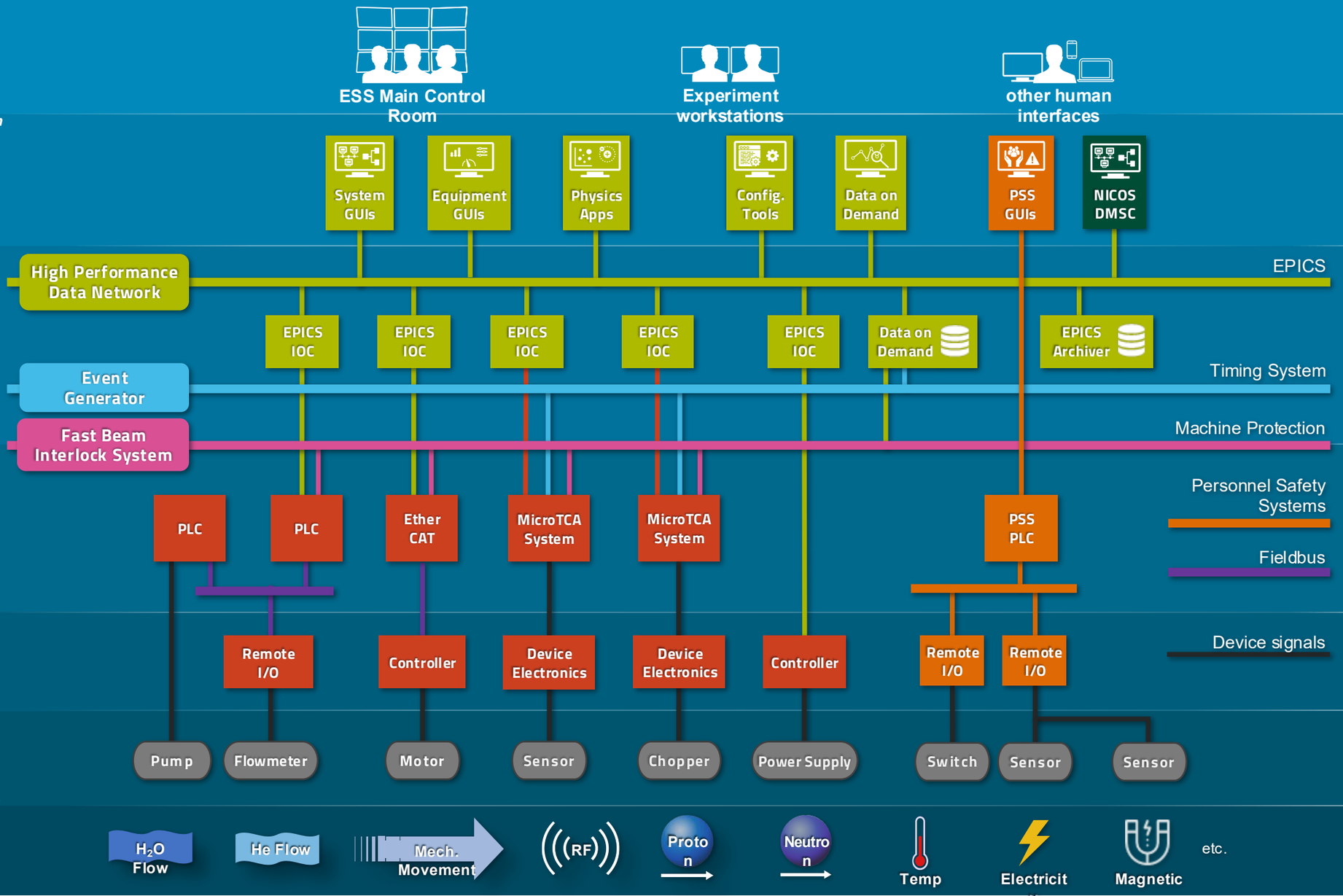
device specific signals

Physical Equipment

machinery, sensors, actuators

measure and adjust

Physical Processes



ESS and EPICS

The newest big EPICS facility, pushing the toolkit forward

Scale — >2700 IOCs and 1.6M archived signals. That makes ESS one of the largest EPICS control systems ever built, and it was designed with EPICS from day one.

E3 — The ESS EPICS Environment: our standard way to build, package and deploy EPICS. It's open, and other labs have adopted ideas from it.

Upstream — ESS developers contribute to EPICS core, Phoebus and the Archiver Appliance, and help maintain them, rather than keeping private forks.

Community — ESS takes part in the EPICS and CS-Studio collaboration meetings and shares what it learns from running at this scale.

Why it matters here — Like the grids in this room, this is federated infrastructure. Many institutions run one shared code base, and none of them owns it alone.

Operations Data Steering Board

Who decides what is recorded, at what rate, and for how long

Stewardship of an archive that outlives every project that fills it

- Formal directives keep signals for the operating lifetime of the facility
- Retention is a commitment, not a storage setting

Reviewing and rationalising what the machine records

- Which signals, at which rates, and which are recorded because nobody has asked to stop

A route for new use cases to reach the data they need

- Collaborations and machine-learning projects need defined, documented access
- Without it, every new question becomes a bespoke extraction

Operations data runs the machine

Different systems, each doing its own job every day

Faults are tracked

Fault tracker · PostgreSQL
Operators log faults during the shift; FBIS injects some automatically. Part of the shift routine, not paperwork afterwards.

Protection events are reconstructable

Synchronous Data Service
When the beam is cut, the whole machine at that instant is captured as one synchronised snapshot.

The machine's history is on record

EPICS archiver · 1.6M PVs
Engineers look back at any signal over days or years: find drift, compare runs, diagnose problems.

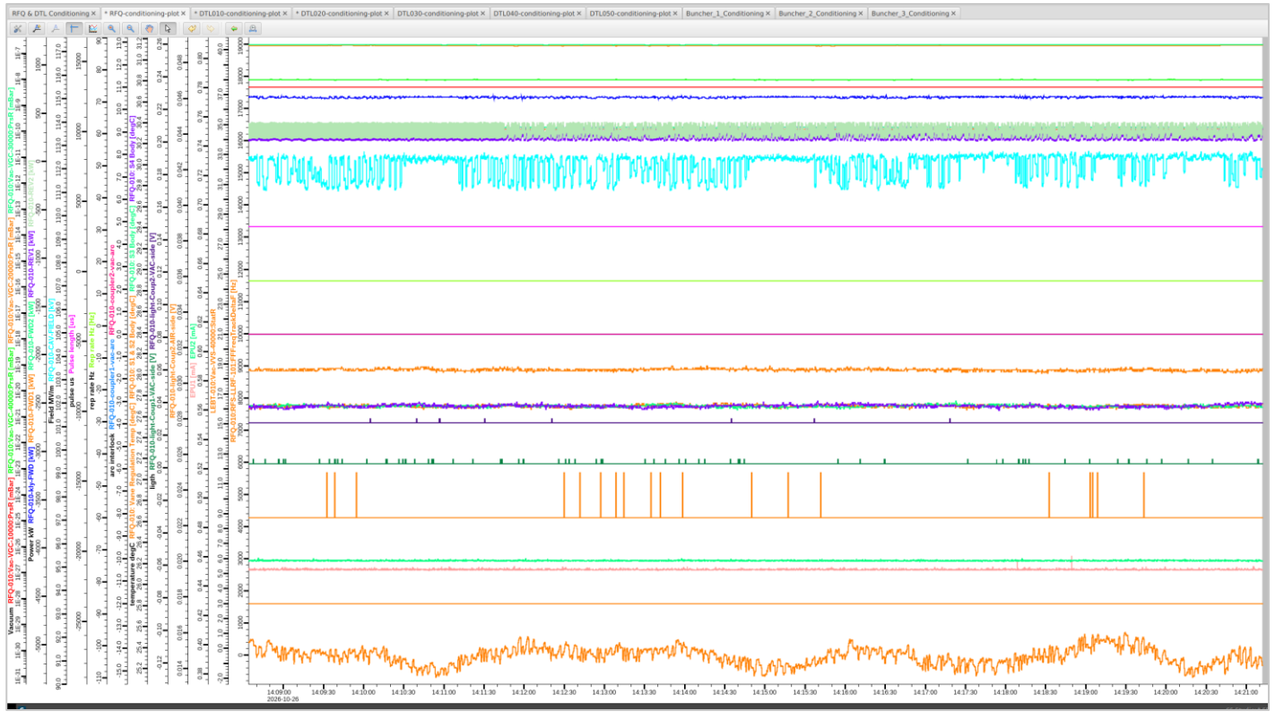
The beam is being commissioned

OpenXAL + Jupyter
Physicists measure beam profiles with OpenXAL, then manage the data themselves and analyse it in notebooks.

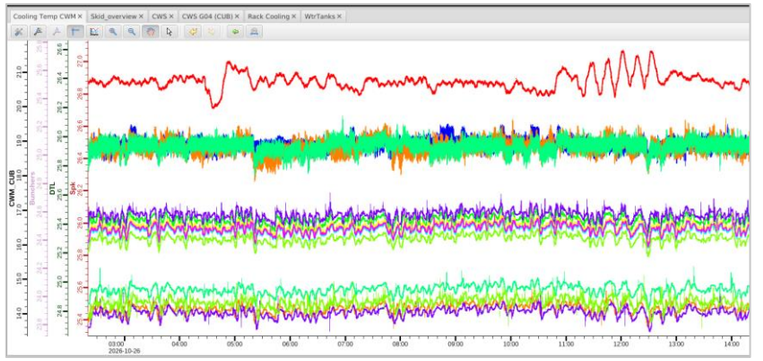
Each works well, each on its own. None of it is machine learning.

Operations Data in action

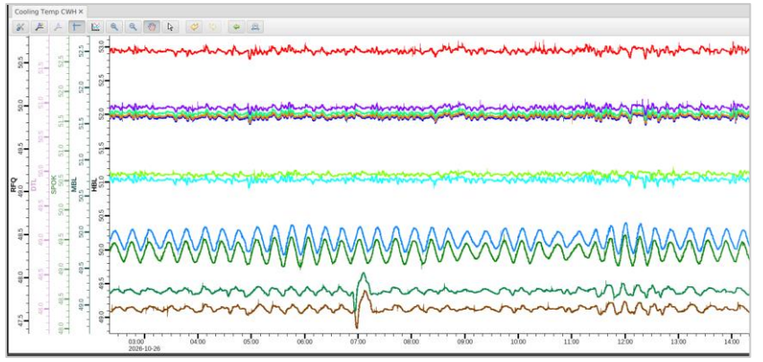
Data Browser combining monitoring and historical data



RFQ conditioning view · 12 minutes, 26 Oct
RF power and field, vacuum, arc and light detectors, temperatures and frequency tracking for RFQ-010 on one time axis.



Cooling water CWM · 12 h
Spoke, DTL, buncher and CUB circuits



Cooling water CWH · 12 h
RFQ, DTL, Spoke, MBL and HBL circuits

Live machine signals next to hours of history: how operators chase the RFQ trips.

Operations Data in action

SDS and Jupyter

Data retrieval with ops-data-client

Import modules

```
[1]: from ops_data_client import get_pvs_by_event_time, get_pvs_by_max_events_per_pv, ConfigureDataSources
     import datetime
```

Configure data sources

In this example, one archiver source and one sds source - there can be several.

```
[2]: config = ConfigureDataSources()
     config.archiver_source("archiver-linac-01.tn.esss.lu.se")
     config.sds_source("sds-db01.tn.esss.lu.se", user="sds_readonly", database="sds")
```

Retrieve data

From PVs known to be in each data source, during a specifed time range.

```
[3]: pv_from_sds = "RFQ-010:RFS-Cav-110:Fld-Wave_SDS-DODData"
     pv_from_archiver = "CrS-PHS:Cryo-D-0100:AmbientTemp"

     df = get_pvs_by_event_time(
         pvs=[pv_from_sds, pv_from_archiver],
         start=datetime.datetime(2025, 6, 18, 4, 57, 35, tzinfo=datetime.timezone.utc),
         end=datetime.datetime(2025, 6, 18, 4, 57, 36, tzinfo=datetime.timezone.utc),
         config=config,
     )
```

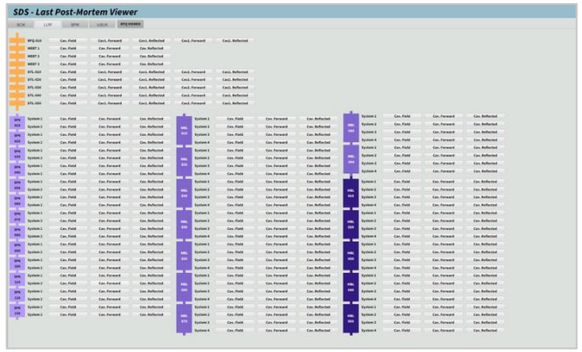
```
[4]: df
```

```
[4]: shape: (17, 8)
```

cycle_id	cycle_ts	pv	data_ts	data_Float64	event_id	event_ts	data_List_Float32
i64	datetime[ns, UTC]	str	datetime[ns, UTC]	f64	u64	datetime[ns, UTC]	list[f32]
15667114431	2025-06-18 04:57:34.540612589 UTC	"RFQ-010:RFS-Cav-110:Fld-Wave_S…	2025-06-18 04:57:34.540612608 UTC	null	15667114443	2025-06-18 04:57:35.412678656 UTC	[4.928, 5.028, … 5.008]
15667114432	2025-06-18 04:57:34.612041157 UTC	"RFQ-010:RFS-Cav-110:Fld-Wave_S…	2025-06-18 04:57:34.612041216 UTC	null	15667114443	2025-06-18 04:57:35.412678656 UTC	[4.888, 4.948, … 4.908]
15667114433	2025-06-18 04:57:34.683469725 UTC	"RFQ-010:RFS-Cav-110:Fld-Wave_S…	2025-06-18 04:57:34.683469824 UTC	null	15667114443	2025-06-18 04:57:35.412678656 UTC	[4.968, 4.948, … 4.818]
15667114434	2025-06-18 04:57:34.754898293 UTC	"RFQ-010:RFS-Cav-110:Fld-Wave_S…	2025-06-18 04:57:34.754898176 UTC	null	15667114443	2025-06-18 04:57:35.412678656 UTC	[5.028, 5.008, … 4.978]
15667114435	2025-06-18 04:57:34.826326861 UTC	"RFQ-010:RFS-Cav-110:Fld-Wave_S…	2025-06-18 04:57:34.826326784 UTC	null	15667114443	2025-06-18 04:57:35.412678656 UTC	[4.968, 4.868, … 4.838]
…	…	…	…	…	…	…	…
15667114442	2025-06-18 04:57:35.326326839 UTC	"RFQ-010:RFS-Cav-110:Fld-Wave_S…	2025-06-18 04:57:35.326326784 UTC	null	15667114443	2025-06-18 04:57:35.412678656 UTC	[5.008, 4.978, … 4.978]
15667114442	2025-06-18 04:57:35.326326839 UTC	"CrS-PHS:Cryo-D-0100:AmbientTem…	2025-06-18 04:57:35.358367745 UTC	16.148724	null	null	null
15667114443	2025-06-18 04:57:35.397755407 UTC	"RFQ-010:RFS-Cav-110:Fld-Wave_S…	2025-06-18 04:57:35.397755392 UTC	null	15667114443	2025-06-18 04:57:35.412678656 UTC	[5.028, 4.978, … 5.008]
15667114444	2025-06-18 04:57:35.469183975 UTC	"RFQ-010:RFS-Cav-110:Fld-Wave_S…	2025-06-18 04:57:35.469184 UTC	null	15667114443	2025-06-18 04:57:35.412678656 UTC	[4.898, 4.958, … 4.958]
15667114445	2025-06-18 04:57:35.540612543 UTC	"CrS-PHS:Cryo-D-0100:AmbientTem…	2025-06-18 04:57:35.559695919 UTC	16.154514	null	null	null

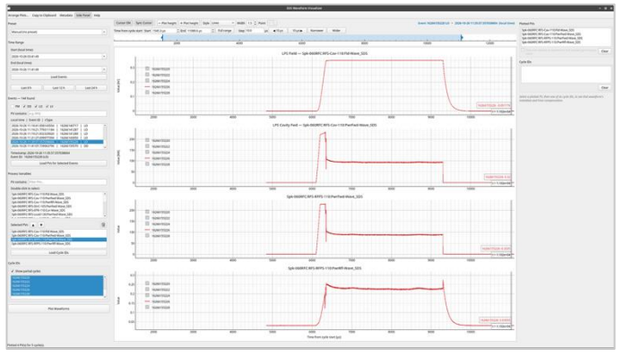
③ ops-data-client in Jupyter

One call, two stores: the RFQ field waveform from SDS (8000 samples per 14 Hz cycle) and an archived temperature, aligned by cycle ID.



① Post-mortem viewer

Last trip, every cavity, RFQ to HBL



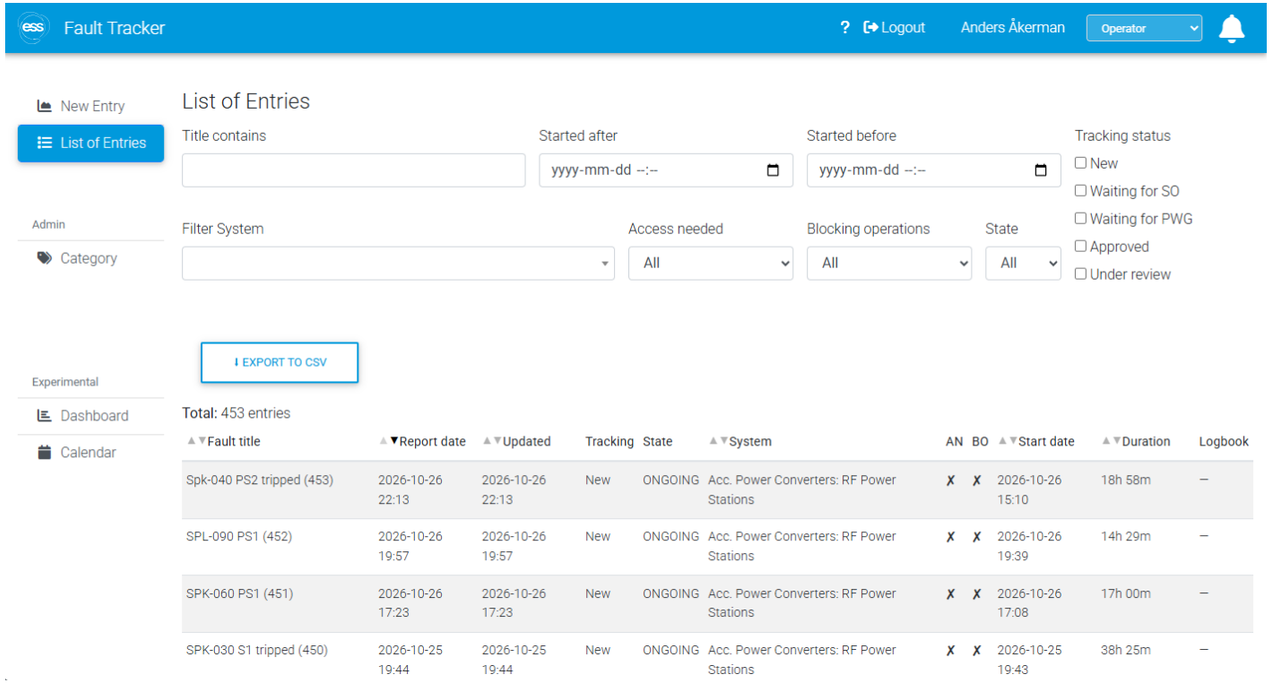
② SDS waveform visualizer

Field, forward and reflected power, cycles around an event

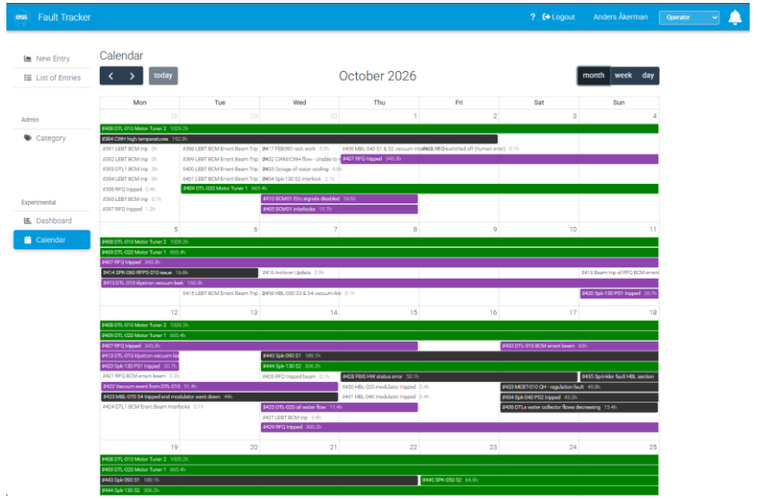
From a trip, to its waveforms, to a dataframe ready for anomaly search.

Operations Data in action

Fault Tracker



List of entries · 453 faults logged
Each fault recorded during the shift with system, state, start and duration; filter by access needed or blocking operations, export to CSV.

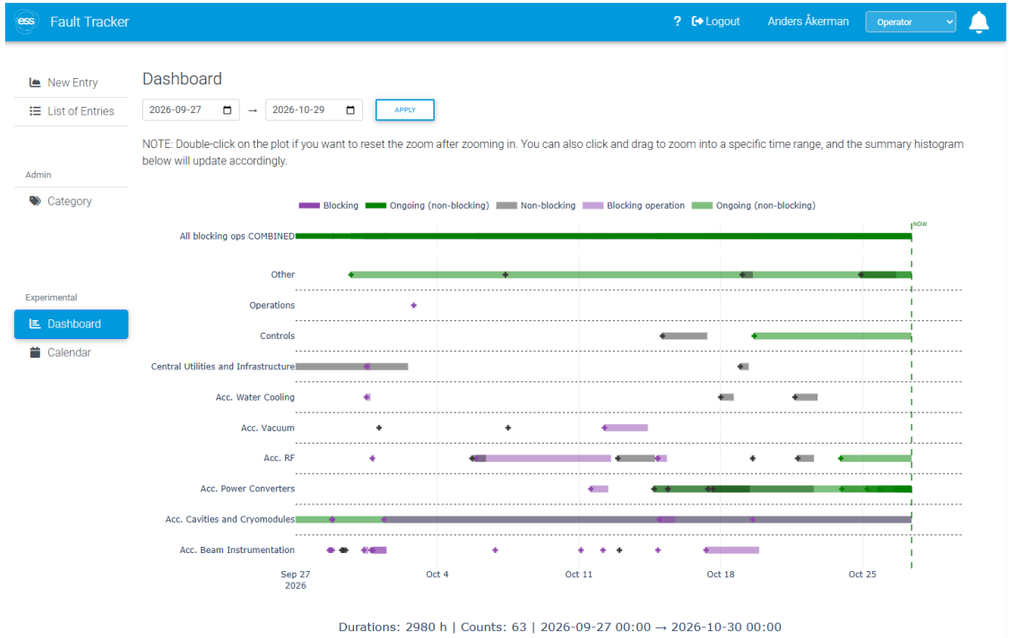


Calendar · October 2026
Every fault drawn across the days it lasted, so the long-running ones stand out.

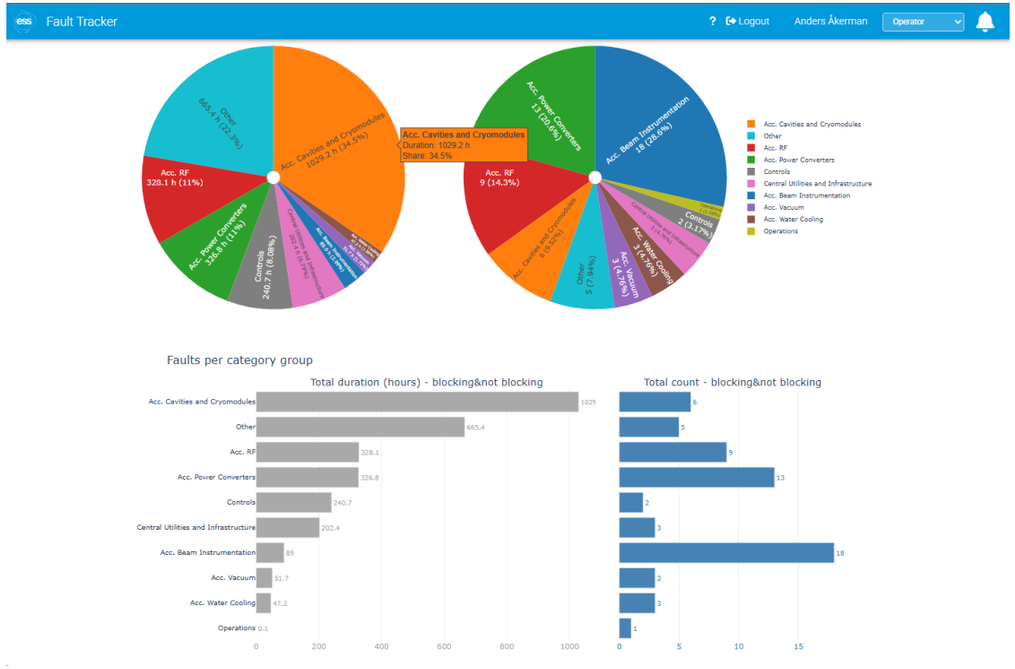
Faults written down as they happen, not reconstructed afterwards.

Operations Data in action

Fault Tracker dashboard



Faults per system · 27 Sep – 26 Oct
63 faults, 2980 hours, blocking and non-blocking on one time axis.



Where the hours go
Cavities and cryomodules: 6 faults, 1029 h. Beam instrumentation: 18 faults, 89 h.

Count and duration tell different stories, and the log is already labelled data.

Who Builds the Tools

None of it is ours alone: open source, built across labs

EPICS — Born at Los Alamos and Argonne around 1990, now an open-source collaboration across 100+ facilities worldwide. ESS is one of its largest installations and is leading the way on its development.

CS-Studio / Phoebus — Started at DESY, carried forward by SNS (Oak Ridge), Brookhaven, ESS and others. The operator screens, alarms, logbook and save & restore you saw run on it.

Archiver Appliance — Created at SLAC with SNS and Brookhaven. ESS runs one of the biggest deployments and contributes back.

OpenXAL — Grew out of SNS's XAL. Now developed jointly by SNS, ESS and other accelerator labs.

The pattern — We take, we fix, we give back. A problem solved at ESS ships to every other lab running the same code.

What the facility asks of the data

Use cases set by the Operations Data Steering Board with the system experts

Use case	What it asks of the data
Facility post-mortem analysis	Synchronised pulse-by-pulse capture across systems, read at three depths: operations, beam physics, subsystem expert
Events of special interest	Collection triggered by conditions experts define — RF field emission, target wheel vibration
Periodical monitoring	Long-term trends from standing collection tasks, for machine studies and availability
Lifecycle and inventory management	Radiation damage, thermo-mechanical fatigue and activation calculations over the facility lifetime
Operational and maintenance decisions	Activation levels of a component before people work on it
Machine and instrument exchange	Machine state during each experiment, and instrument state back to the machine
AI/ML applications	Anomaly detection, fault diagnosis, predictive maintenance, validating changes against a baseline

Every one of these needs data from several systems at once, and from more than one store.

From reactive towards predictive

What the field has shown an archive like this can support

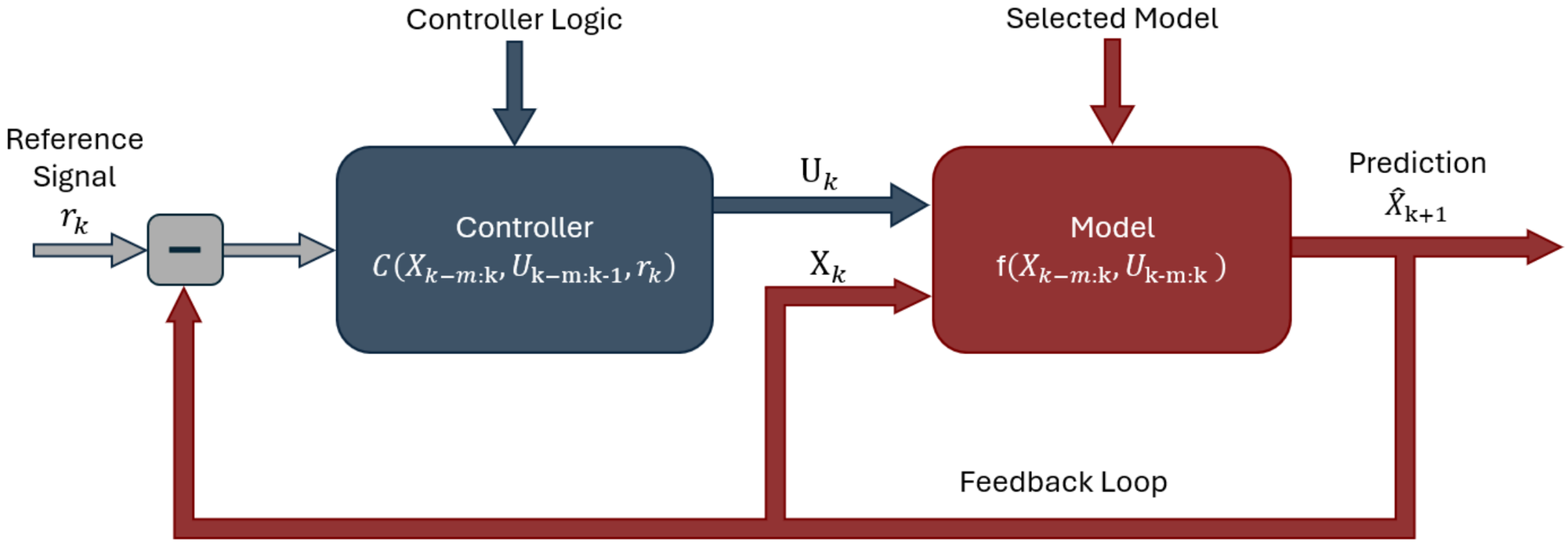
Where we are

Reactive	Classified	Predictive	Autonomous
Fix it after the beam is lost. Logs read by people who already know what to look for.	Faults grouped automatically across the archive, so recurring failure modes surface on their own.	Errant beam flagged before it is lost, drift caught before it trips protection.	The machine holds its own operating point, with people setting intent rather than parameters.

Other facilities have already demonstrated the middle two steps. At ESS the interest is organised — research collaborations with universities and industry, and a growing set of machine-learning projects working against control-system data — but almost none of it has yet been done with this archive as a whole.

Learning a plant too complex to model

Transformer and LSTM models of the target helium cooling system, trained on archived operations data

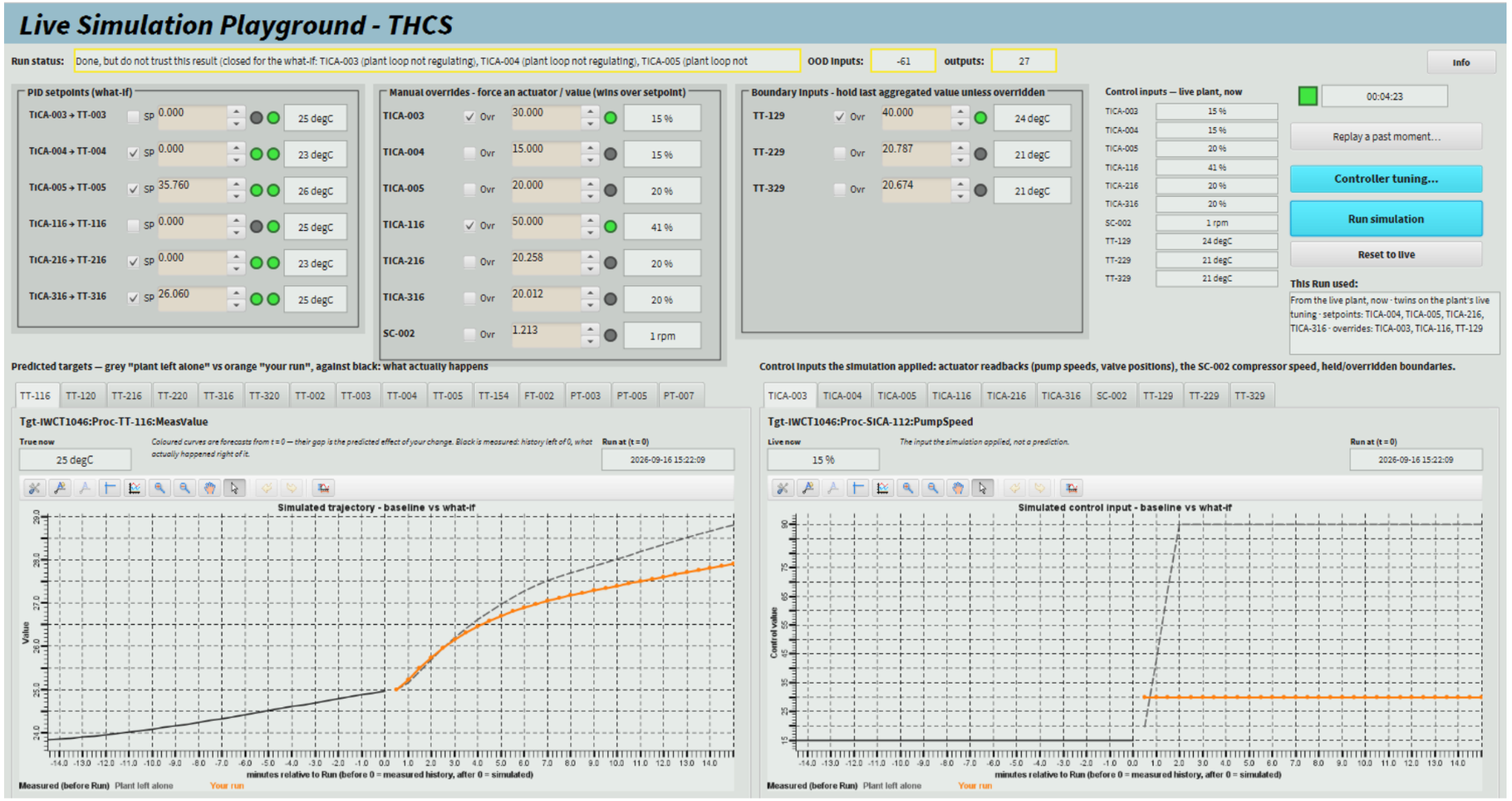


Control variables go in as inputs, so the model answers what happens if we do this. Each prediction feeds the next, in training and in use, so the loop closes.

The helium cooling system predicted well; the cryogenic moderator system did not — too little archived data to learn from.

A what-if you can run on the live plant

The simulation starts from the plant as it is now, or from any moment in the archive



Grey is the plant left alone, orange the operator's change, black what actually happened. Training the model and judging the forecast both need the archive.

Collaboration and data sharing

NOW

Operations data sharing policy

- A formal route for external access to ESS operations data — about to be signed
- First users: university student machine learning projects
- Anomaly detection, fault prediction, beam tuning studies

POSSIBLY NEXT

Heavier AI workloads

- If student projects show value, models may outgrow a laptop
- Training on long archive histories would need real compute
- Not on the roadmap yet — an open question, not a plan

THE GAP

EPICS tools are not built for ML

- Archiver, PV Access and logbook APIs serve operators, not bulk extraction
- Data comes out PV by PV — no columnar formats, no ML-ready datasets
- No catalogue or FAIR layer for outside users

An open question for this community: how have others made control-system data ML-ready — and shared it?

Thank you

This talk is their work; any mistakes in it are mine

ICS Software Group — archiver, SDS, Fault Tracker, OpenXAL

- Abraham Wolk, Anders Åkerman, Balazs Nagy, Darren Spruce, Dirk Nordt, Georg Weiss, Jason Brudvik, Marco Leorato, Max Frederiksen, Ricardo Fernandes, Simon Rose, Sky Brewerand everyone who built the control system software before 2024

IT Infrastructure & Operations — the compute, storage and network behind the archive

- Stephane Almanet — zoning, CEPH and infrastructure diagrams

Operations, physics and research

- Control Room Operations — Iordanis Karagiannis, for showing the current operator plots
- Fault Tracker – Anders Åkerman for the current screenshots.
- Christopher Källström, Lund University — Data Quality and Quantity for ML at ESS (2025)



Thankyou