SOLARIS
CENTRE

SOLARIS Status Update

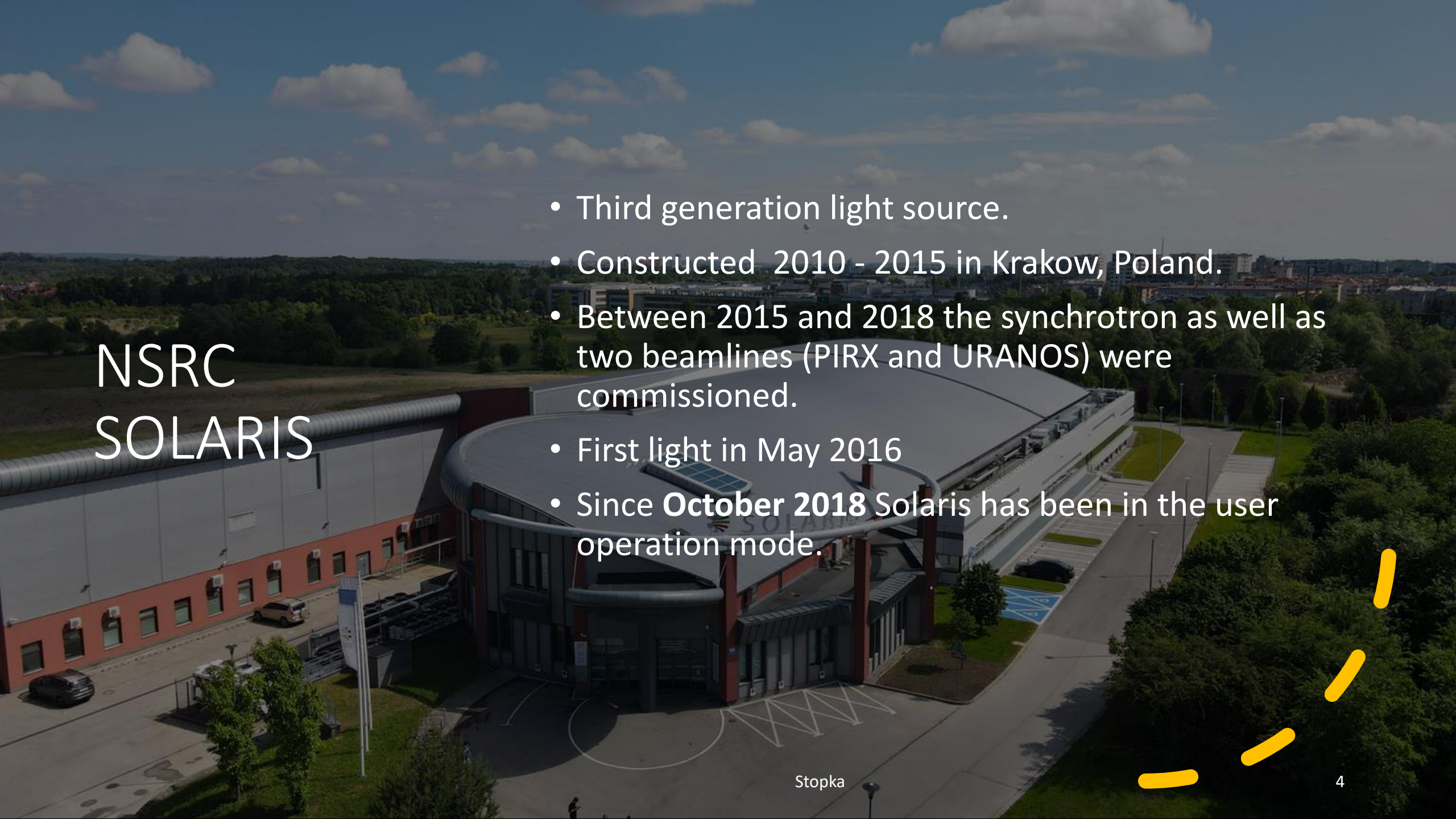
Cerdanyola del Vallès, 40th Tango Community Meeting, 8-12.06.2026

Michał Piekarski, Urszula Jachymczyk, SOLARIS CSIT Team
SOLARIS National Synchrotron Radiation Centre

NSRC SOLARIS

Fot. Aleksander Koczur

NSRC SOLARIS



- Third generation light source.
- Constructed 2010 - 2015 in Krakow, Poland.
- Between 2015 and 2018 the synchrotron as well as two beamlines (PIRX and URANOS) were commissioned.
- First light in May 2016
- Since **October 2018** Solaris has been in the user operation mode.

SOLARIS research infrastructure

Available and in user operation

Starting up

Under construction

Project application accepted

Available space for 4-5 beamlines

+

CryoEm centre

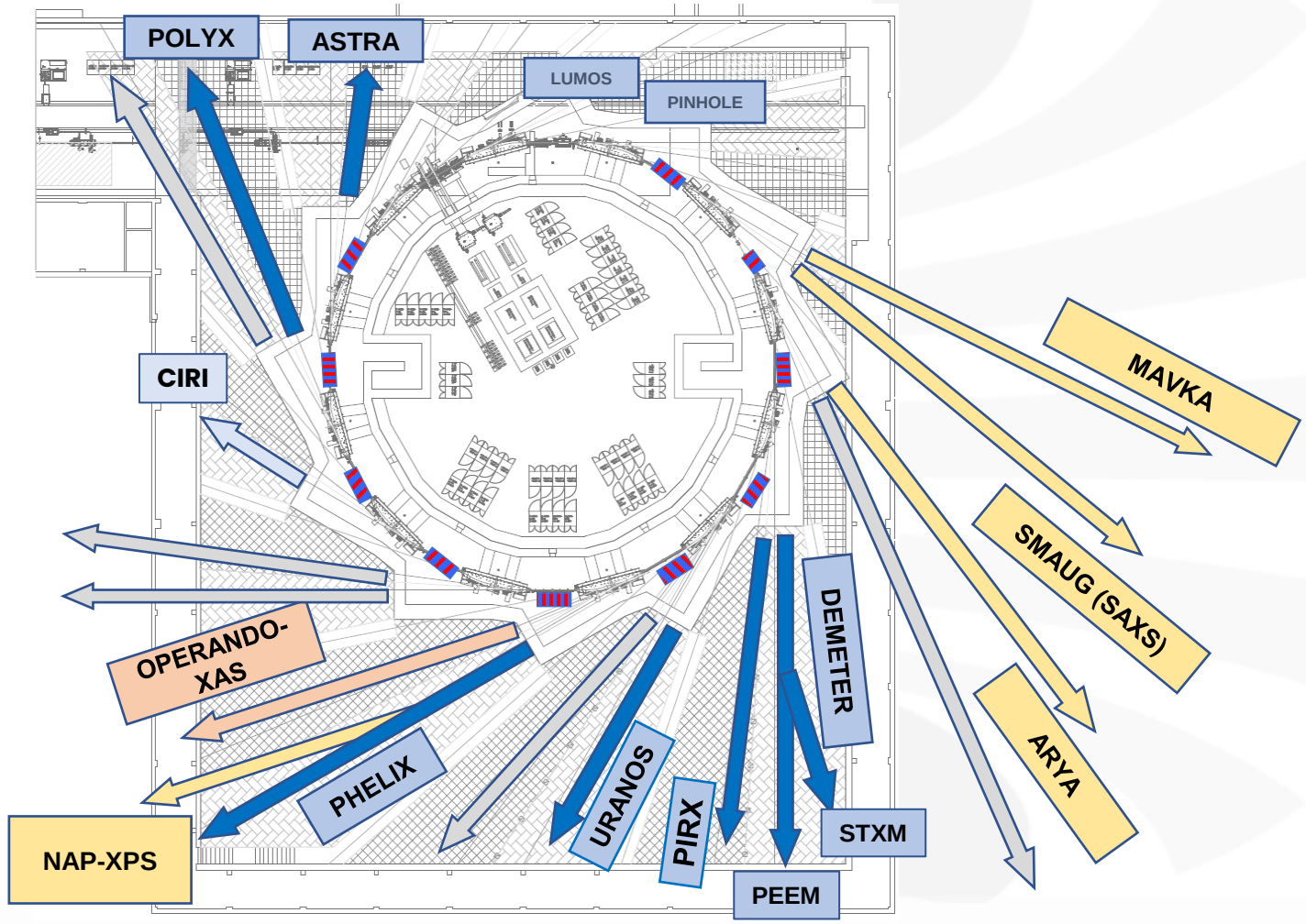
Structural Biology Core Facility

Krios

Glacios

Neutron-lab

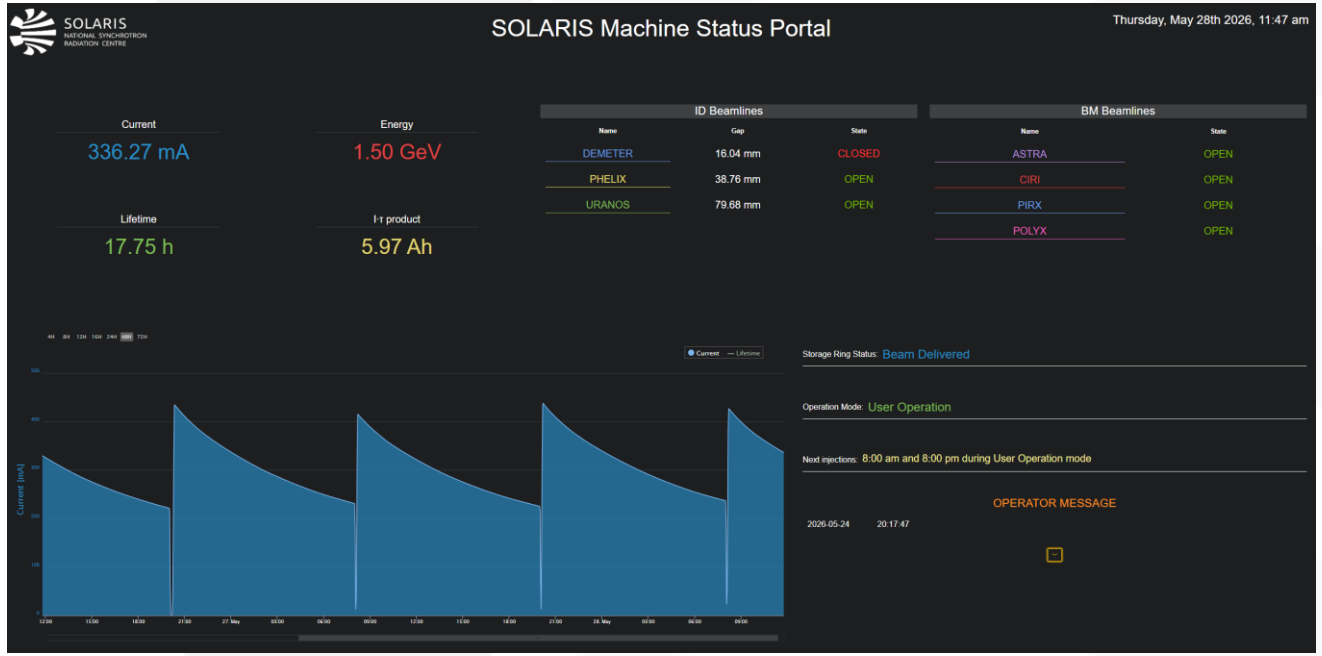
Research University Excellence Initiative



jakub.szlachetko@uj.edu.pl

SOLARIS Operation

- Operation in the decay mode
- Injection twice/day: 8:00 am and 8 pm
- Monday – machine days, maintenance
- User operation 6 days/week (Tue-Sun)
- One operation mode (uniform filling pattern)
- Singe bunch operation mode under development



2026

SOLARIS CSIT Department today

- 24 people in CSIT
  - 8 Control Systems,
  - 7 IT
    - 3 pure IT/helpdesk,
    - 1 infra admin,
    - 1 cyber-security + networks,
    - 2 DevOps,
  - 7 Web-SUN / CS

Control System upgrade – achievements so far and goal for 2026

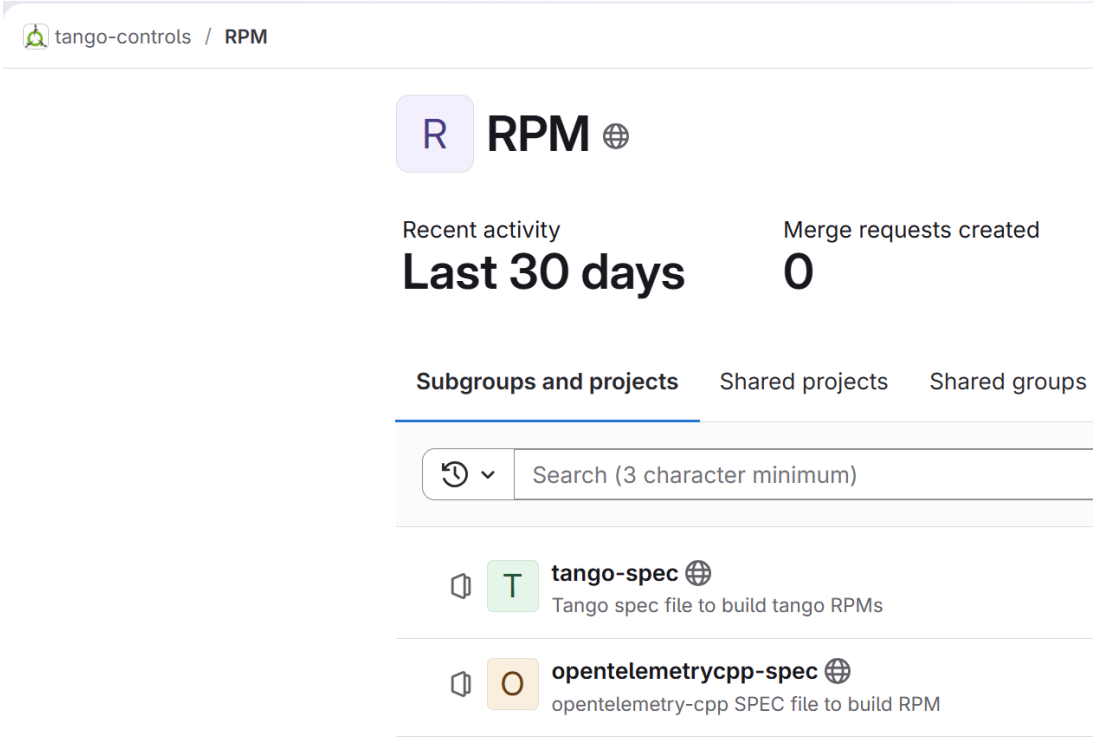
- AWX/Ansible updated to the latest version (Execution Enviroments) and moved to the Kubernetes cluster
- Prepared CI/CD to support two OS (el9 and el10) – for internal and external projects.
- Main Tango packages and dependencies have been built (including Tango10/pyTango10)
- All workstations in CR and on the beamlines are migrated to Alma 9
  - MATE not yet officially available for AlmaLinux10
- All of the GUIs migrated to at least Python 3.9 and Taurus 5 – even these in Python 2.7 and Taurus 3.7 version
- All VMs on the beamlines migrated to Alma9
- **GOAL: Migrate all hosts to at least Alma 9 (some already to Alma10),**
  - Summer shutdown: machine VMs migration (27 VMs, ~100 projects)
    - Alma10
    - Tango10

Control System stack in NSRC SOLARIS: summary

- OS: **Alma 9 (mainly) and 10,** Windows 10, 11 (mainly) + some embedded custom Linux
- Python versions: 3.8 (some web), **3.9 (mainly) and 3.12**
- Tango versions: **9.3.6 (mainly)**, 9.2 (embedded) -> **soon 10.3.0**
- Taurus versions: **5.1 (mainly)**, -> **soon 5.4.0**
- Sardana versions: **3.6 only** -> **soon 3.7**
- Additionally, WEB projects use Vue.js, Tango GQL and Taranta

- Cryomicroscopes are fully independent and are based on manufacturer software

SOLARIS took over Tango RPMs



Results

Source state: succeeded
Source build logs: builder-live.log.gz , backend.log.gz
Built Packages:
tango-common 10.3.0
tango-starter-debuginfo 9.0
tango-db-debuginfo 5.28
tango-idl 6.0.4
tango-debuginfo 10.3.0
libtango10-debuginfo 10.3.0
tango-accesscontrol 2.22
libtango10 10.3.0
tango-test 3.13
tango-debugsource 10.3.0
tango-starter 9.0
tango-universaltest-debuginfo 1.4
tango-accesscontrol-debuginfo 2.22
tango-universaltest 1.4
libtango10-devel 10.3.0
tango-admin 1.28
tango-admin-debuginfo 1.28
tango-rest-server 1.22
tango-java 10.3.0
tango-test-debuginfo 3.13
tango-db 5.28

fedora copr

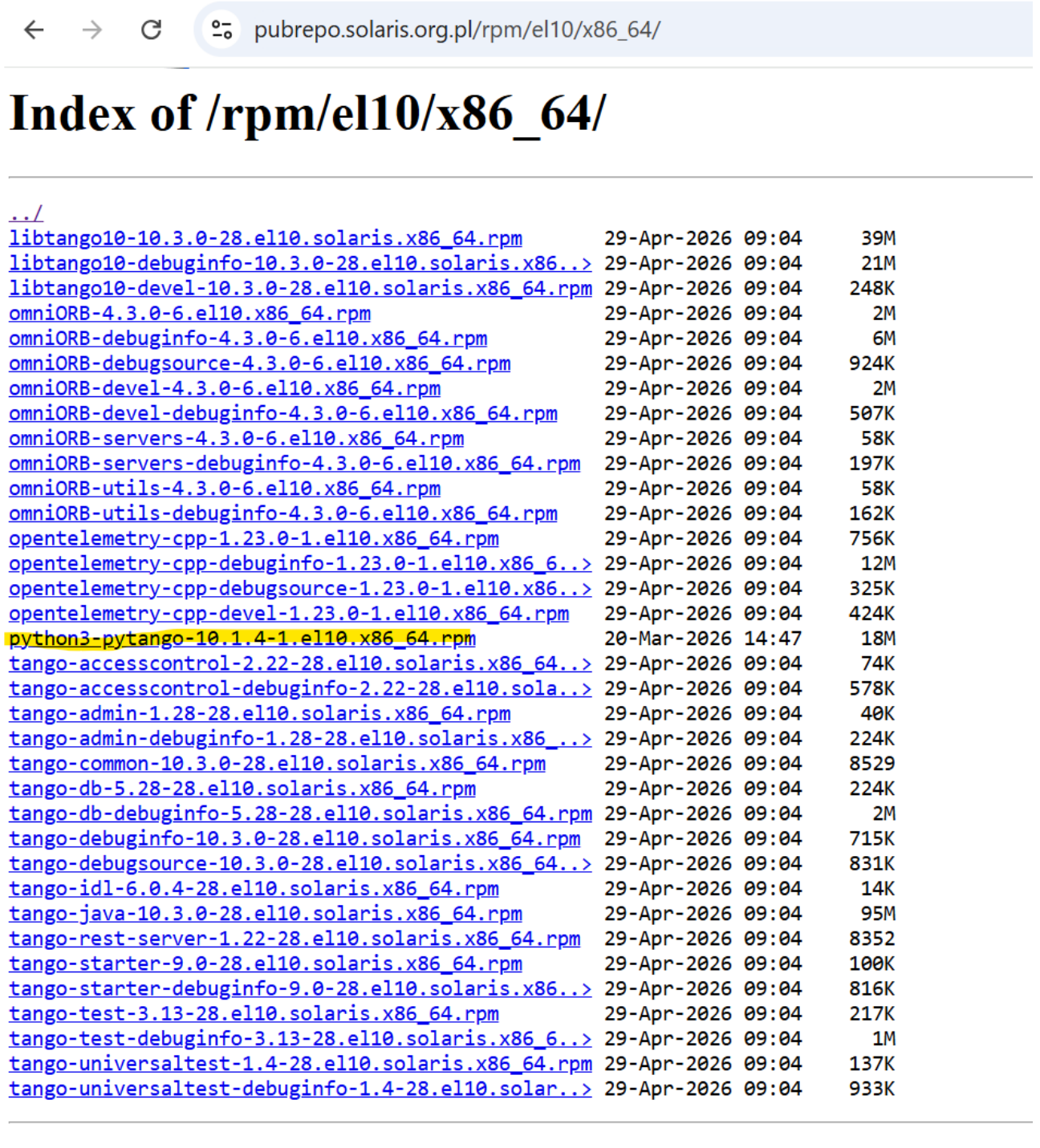
Chroot Name	Dist Git Source	Build Time
epel-10-x86_64	b965f0d	8 minutes
epel-9-x86_64	b965f0d	8 minutes

PyTango

fpm - packaging made simple
latest

- tricky (with build deps, toml…)
- used FPM from wheel to create RPM

https://pubrepo.solaris.org.pl/rpm



Some new developments

Motors Snapshots Sardana toolset

What do we have so far?

- Manages different positions profiles for motors,
- Makes snaps and saves them in dedicated catalog,

List & set profile

```
Door_lab-test_1 [6]: lsprofs
Available profiles:
[1]     FTIR
[2]     OPTIR
[3]     NANO-FTIR
```

```
Door_lab-test_1 [7]: setprof FTIR
Snap profile set to: FTIR
```

Motors Snapshots Sardana toolset

What do we have so far?

- Manages different positions profiles for motors,
- Makes snaps and saves them in dedicated catalog,
- Maintains snapshots' history for selected mode,

List & set profile

```
Door_lab-test_1 [6]: lsprofs
Available profiles:
[1]     FTIR
[2]     OPTIR
[3]     NANO-FTIR
```

```
Door_lab-test_1 [7]: setprof FTIR
Snap profile set to: FTIR
```

Lists available snaps

```
Door_lab-test_1 [8]: lssnaps
Available snapshots for 'FTIR' profile:
[1]     000_2026_05_14_13_22_11.json
[2]     001_2026_05_25_16_29_58.json
[3]     002_2026_05_25_16_38_20.json
[4]     003_2026_05_28_14_59_38.json
```

Motors Snapshots Sardana toolset

What do we have so far?

- Manages different positions profiles for motors,
- Makes snaps and saves them in dedicated catalog,
- Maintains snapshots' history for selected mode,
- Displays data saved in snap,

Display snap file content

```
Door_lab-test_1 [13]: getsnap 2
-----------------------------------------------
SNAP ID:        2
PROFILE:        FTIR
SNAP NAME:      002_2026_05_25_16_38_20.json
SNAP DESC:      FROM ATK ZERO

MOTORS
======

Rotation_L

DIAL POSITION:  0.0
POSITION:       0.0
SIGN:           1
OFFSET:         0.0
BACKLASH:       0.0
STEP PER UNIT:  111.1111111

------------------------

Rotation_R

DIAL POSITION:  0.0
POSITION:       0.0
SIGN:           1
OFFSET:         0.0
BACKLASH:       0.0
STEP PER UNIT:  111.1111111

------------------------
```

List & set profile

```
Door_lab-test_1 [6]: lsprofs
Available profiles:
[1]     FTIR
[2]     OPTIR
[3]     NANO-FTIR
```

```
Door_lab-test_1 [7]: setprof FTIR
Snap profile set to: FTIR
```

Lists available snaps

```
Door_lab-test_1 [8]: lssnaps
Available snapshots for 'FTIR' profile:
[1]     000_2026_05_14_13_22_11.json
[2]     001_2026_05_25_16_29_58.json
[3]     002_2026_05_25_16_38_20.json
[4]     003_2026_05_28_14_59_38.json
```

Motors Snapshots Sardana toolset

What do we have so far?

- Manages different positions profiles for motors,
- Makes snaps and saves them in dedicated catalog,
- Maintains snapshots' history for selected mode,
- Displays data saved in snap,
- Restores motors' config from chosen snap,
- Deletes snap.

Display snap file content

```
Door_lab-test_1 [13]: getsnap 2
-----------------------------------------------
SNAP ID:        2
PROFILE:        FTIR
SNAP NAME:      002_2026_05_25_16_38_20.json
SNAP DESC:      FROM ATK ZERO

MOTORS
======

Rotation_L

DIAL POSITION:  0.0
POSITION:       0.0
SIGN:           1
OFFSET:         0.0
BACKLASH:       0.0
STEP PER UNIT:  111.1111111

------------------------

Rotation_R

DIAL POSITION:  0.0
POSITION:       0.0
SIGN:           1
OFFSET:         0.0
BACKLASH:       0.0
STEP PER UNIT:  111.1111111

------------------------
```

List & set profile

```
Door_lab-test_1 [6]: lsprofs
Available profiles:
[1]     FTIR
[2]     OPTIR
[3]     NANO-FTIR
```

```
Door_lab-test_1 [7]: setprof FTIR
Snap profile set to: FTIR
```

Lists available snaps

```
Door_lab-test_1 [8]: lssnaps
Available snapshots for 'FTIR' profile:
[1]     000_2026_05_14_13_22_11.json
[2]     001_2026_05_25_16_29_58.json
[3]     002_2026_05_25_16_38_20.json
[4]     003_2026_05_28_14_59_38.json
```

Restore motors config from snap

```
Door_lab-test_1 [14]: rstsnap 2
Restoring Rotation_L...
Sign: NO CHANGE
Offset: NO CHANGE
Position: RESTORED

Restoring Rotation_R...
Sign: NO CHANGE
Offset: NO CHANGE
Position: RESTORED

DONE
```

Motors Snapshots Sardana toolset

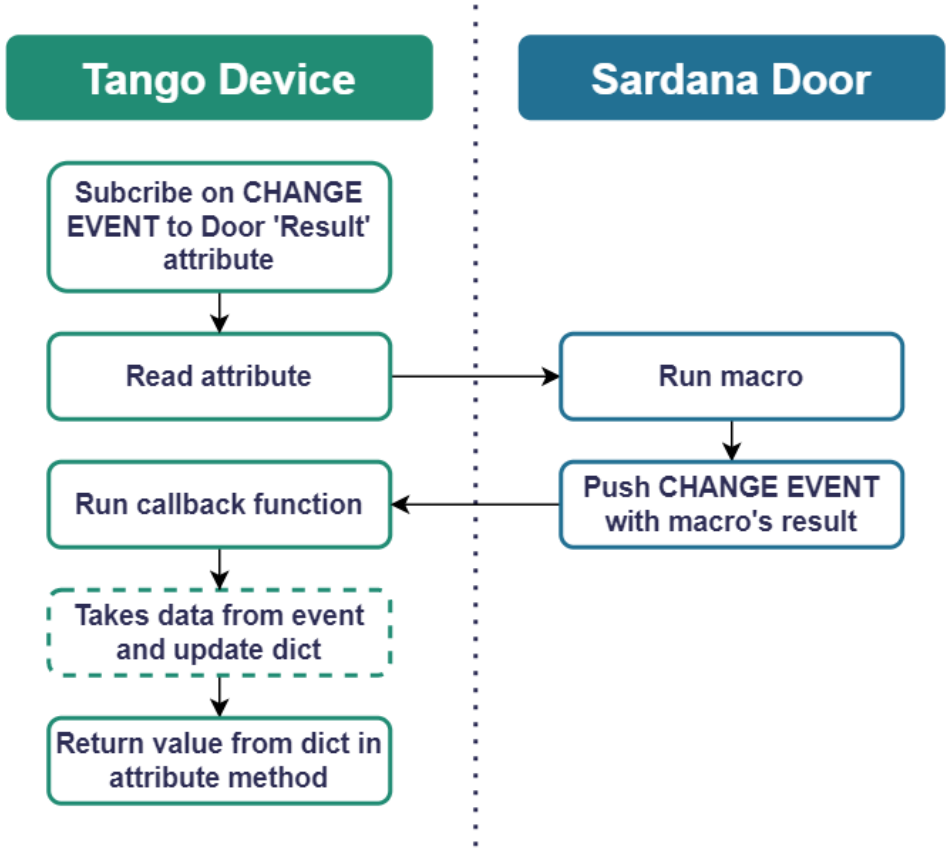
We are working on:

- Logic layer (Tango Device) which implements *Attributes* and *Commands* **based on Macro's result**,
  - Basic attributes – DONE
  - Basic commands – DONE
  - Command that takes more complex input parameters – IN PROGRESS
- GUI build on top of logic layer (tango device) – IN PROGRESS

Attribute 'snaps' - implementation example

```
@attribute(
    dtype=(str,),
    max_dim_x=1024,
    access=AttrWriteType.READ,
    label="Snapshots",
    doc="Available snapshots for currently set profile.",
)
def snaps(self) -> List[str]:
    self._run_macro("lssnaps")
    return self._attrs_dict["snaps"]
```

Workflow details



Banana Collision

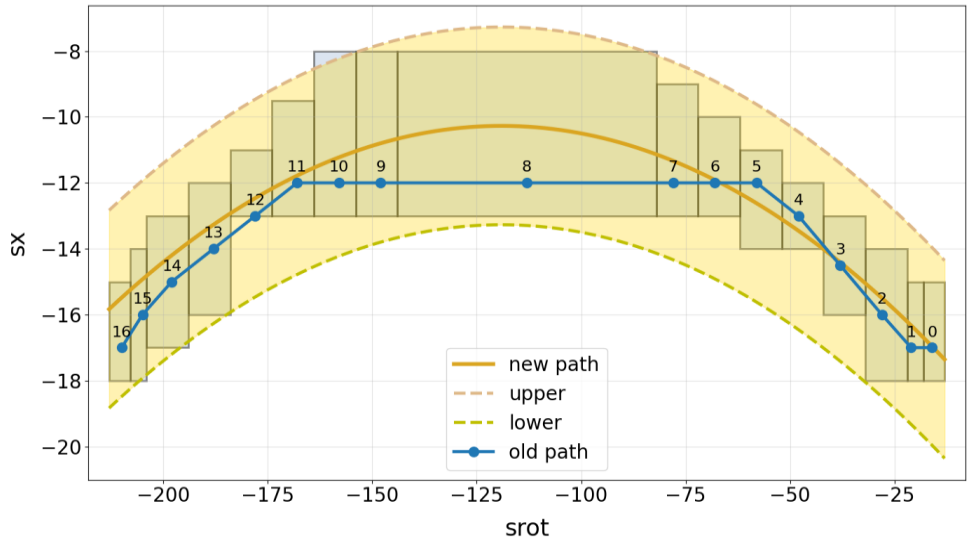
config.json

- motors_proxy
- axes + default ranges
- rules
  - pair_range
  - pair_function_range
    - circle
    - corridor

Banana Collision

config.json

- motors_proxy
- axes + default ranges
- rules
  - pair_range
  - pair_function_range
    - circle
    - corridor

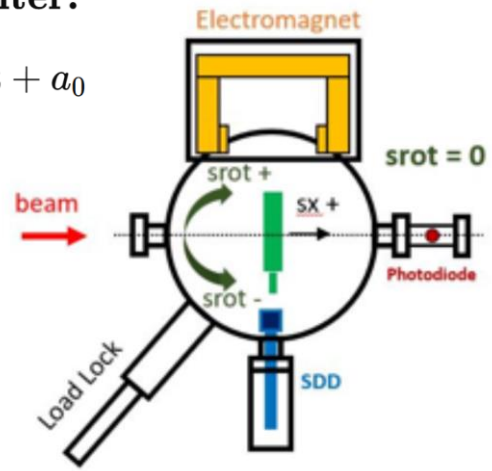


Polynomial corridor center:

$$\mathrm{sx}_c(\mathrm{srot}) = a_2\mathrm{srot}^2 + a_1\mathrm{srot} + a_0$$

Corridor constraint:

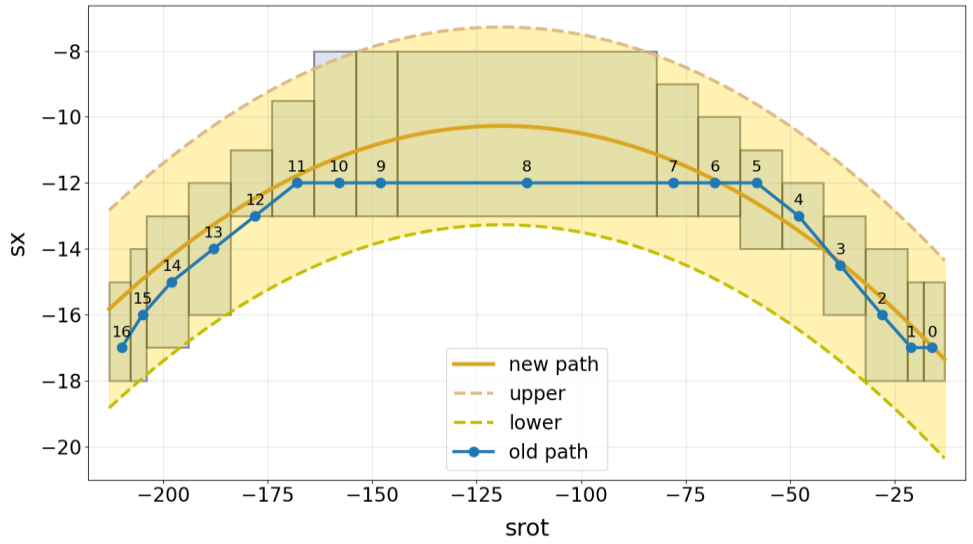
$$|\mathrm{sx} - \mathrm{sx}_c(\mathrm{srot})| \leq h$$



Banana Collision

config.json

- motors_proxy
- axes + default ranges
- rules
  - pair_range
  - pair_function_range
    - circle
    - corridor

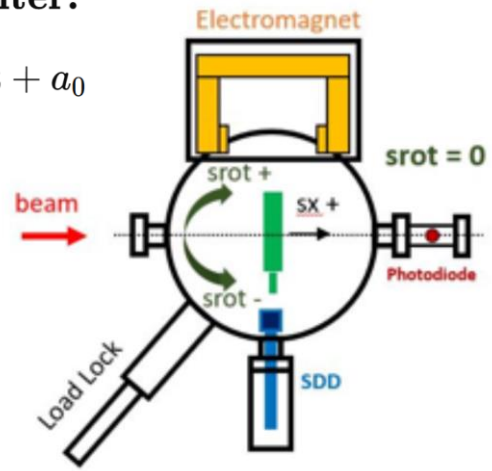


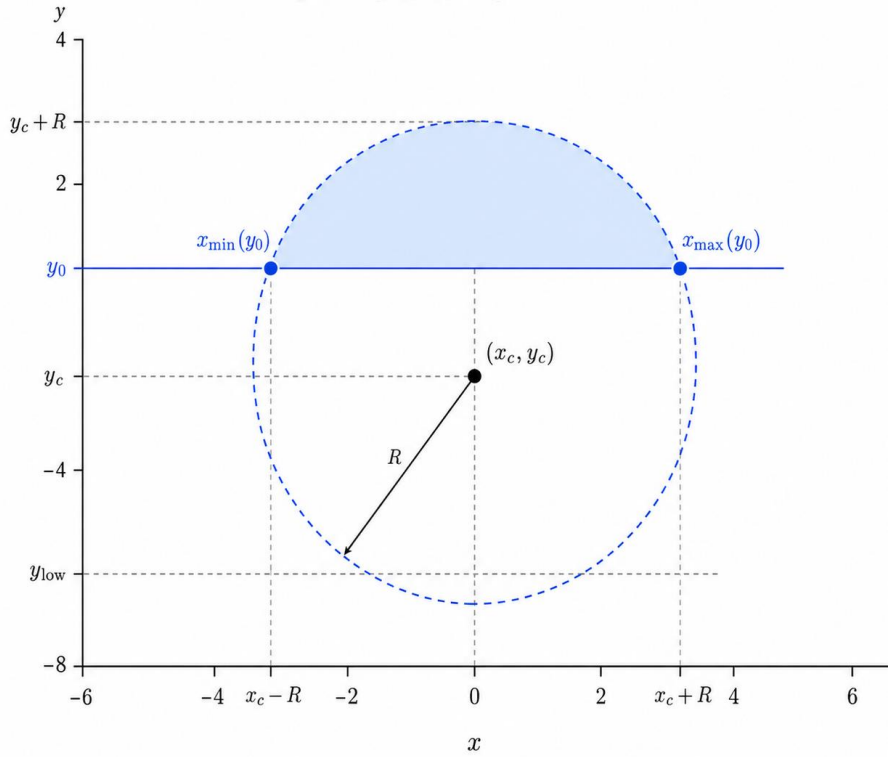
Polynomial corridor center:

$$\mathrm{sx}_c(\mathrm{srot}) = a_2\mathrm{srot}^2 + a_1\mathrm{srot} + a_0$$

Corridor constraint:

$$|\mathrm{sx} - \mathrm{sx}_c(\mathrm{srot})| \leq h$$





Circle equation:

$$(x - x_c)^2 + (y - y_c)^2 \leq R^2$$

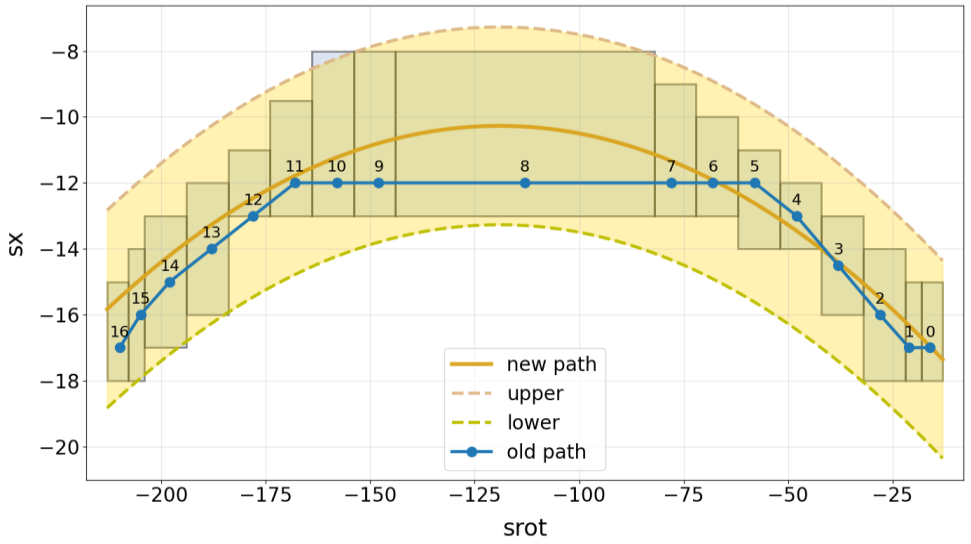
Circle constrains:

$$x \in \left[x_c - \sqrt{R^2 - (y - y_c)^2},\ x_c + \sqrt{R^2 - (y - y_c)^2}\right]$$

Banana Collision

config.json

- motors_proxy
- axes + default ranges
- rules
  - pair_range
  - pair_function_range
    - circle
    - corridor

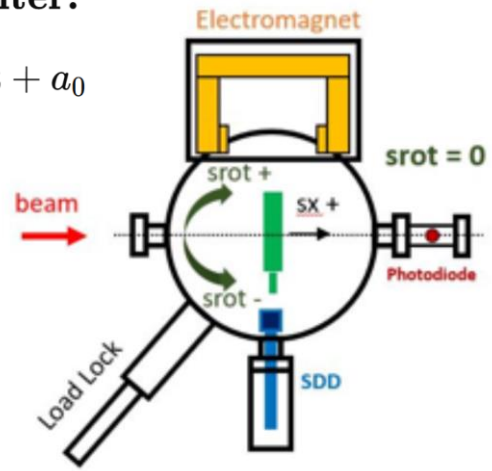


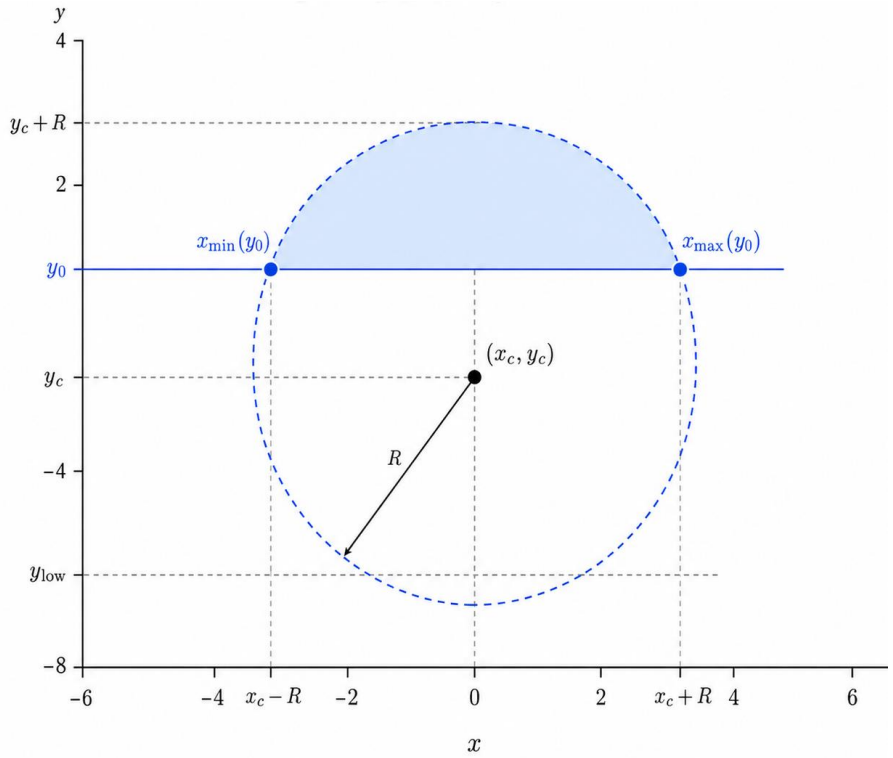
Polynomial corridor center:

$$\mathrm{sx}_c(\mathrm{srot}) = a_2\mathrm{srot}^2 + a_1\mathrm{srot} + a_0$$

Corridor constraint:

$$|\mathrm{sx} - \mathrm{sx}_c(\mathrm{srot})| \le h$$





Circle equation:

$$(x - x_c)^2 + (y - y_c)^2 \le R^2$$

Circle constrains:

$$x \in \left[x_c - \sqrt{R^2 - (y - y_c)^2},\ x_c + \sqrt{R^2 - (y - y_c)^2}\right]$$

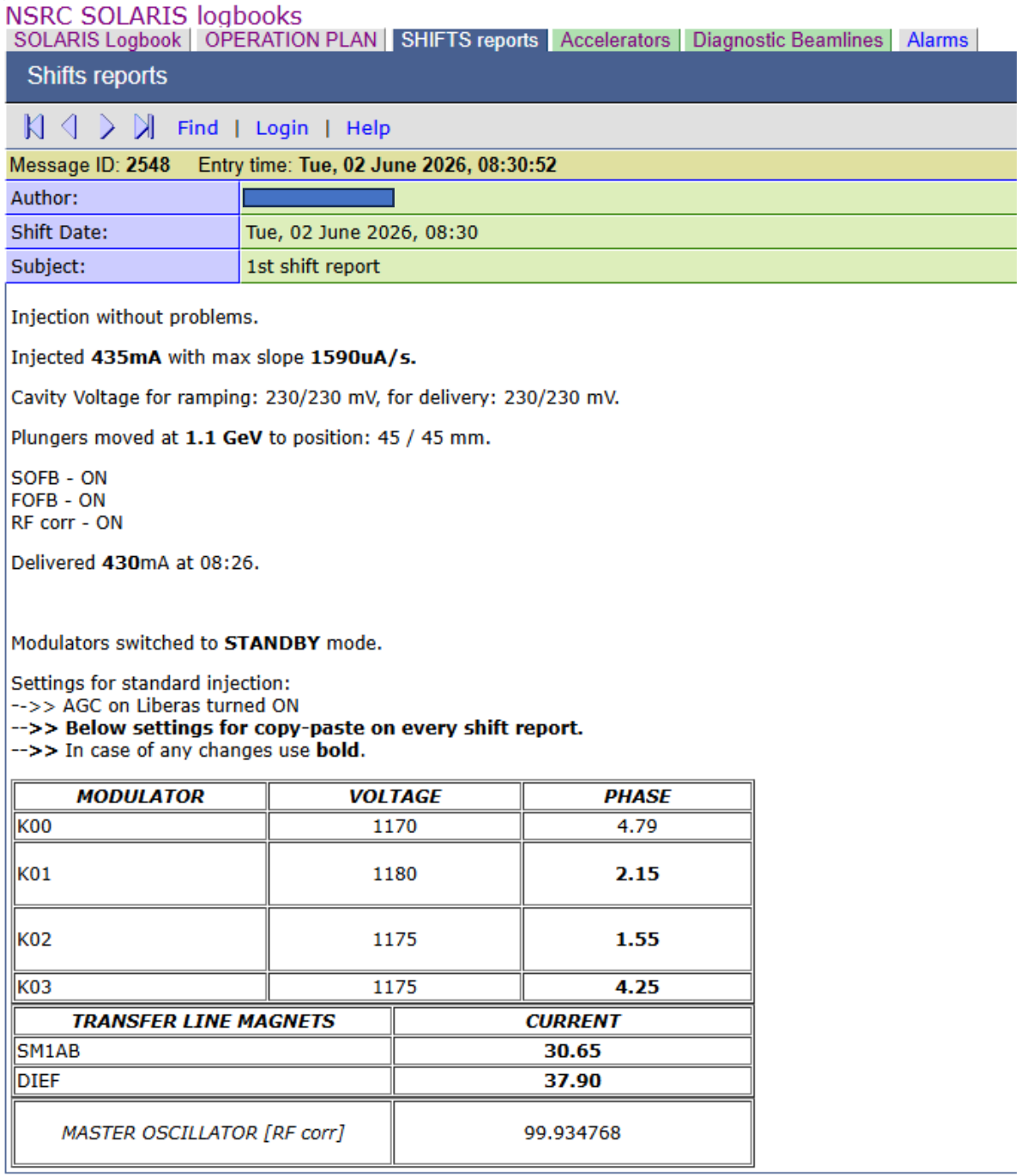
Function limits work both ways:

Sx ⟺ Srot

x ⟺ y

Automatization

- Automatic injection Logbook report
  - the report is created by filling the template with data derived from multiple DSs.
    ~~copy + paste, manual attributes searching~~
- Automatic MSP communication after beam lost
  - After the logical value for beam dump is met, communication is shown at Machine Status Portal → informed users ☺



NSRC SOLARIS logbooks

SOLARIS Logbook | OPERATION PLAN | SHIFTS reports | Accelerators | Diagnostic Beamlines | Alarms

Shifts reports

Find | Login | Help

Message ID: **2548** Entry time: **Tue, 02 June 2026, 08:30:52**

Author:	[redacted]
Shift Date:	Tue, 02 June 2026, 08:30
Subject:	1st shift report

Injection without problems.

Injected **435mA** with max slope **1590uA/s.**

Cavity Voltage for ramping: 230/230 mV, for delivery: 230/230 mV.

Plungers moved at **1.1 GeV** to position: 45 / 45 mm.

SOFB - ON
FOFB - ON
RF corr - ON

Delivered **430**mA at 08:26.

Modulators switched to **STANDBY** mode.

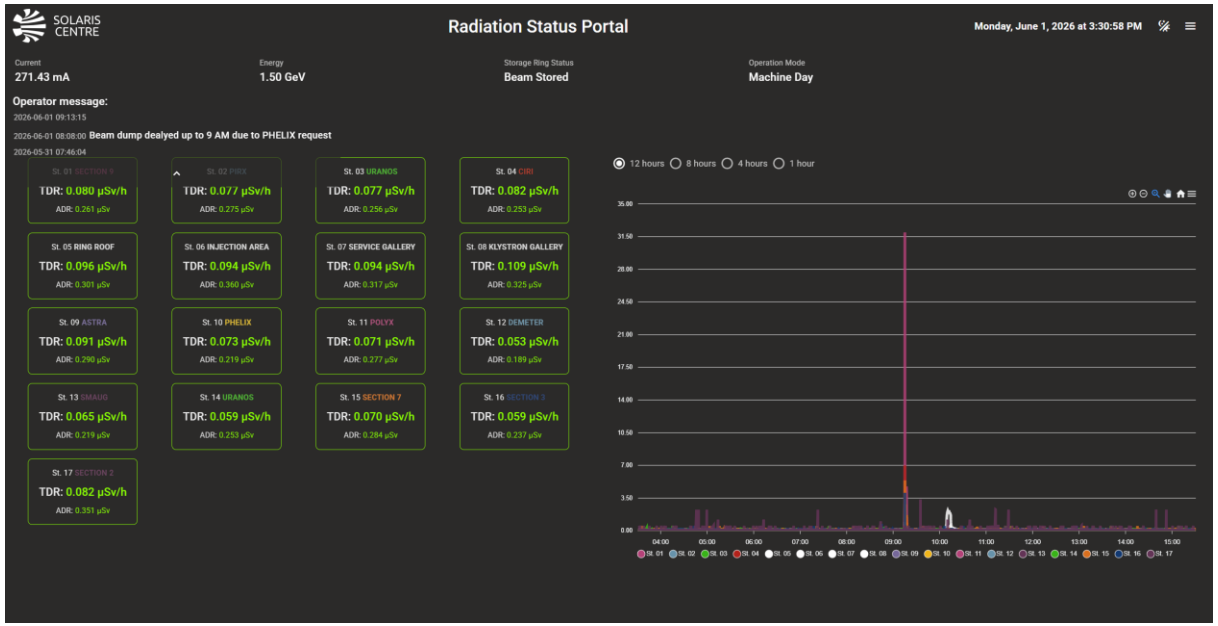
Settings for standard injection:
-->> AGC on Liberas turned ON
-->> Below settings for copy-paste on every shift report.
-->> In case of any changes use **bold**.

MODULATOR	*VOLTAGE*	*PHASE*
K00	1170	4.79
K01	1180	**2.15**
K02	1175	**1.55**
K03	1175	**4.25**

TRANSFER LINE MAGNETS	*CURRENT*
SM1AB	**30.65**
DIEF	**37.90**
MASTER OSCILLATOR [RF corr]	99.934768

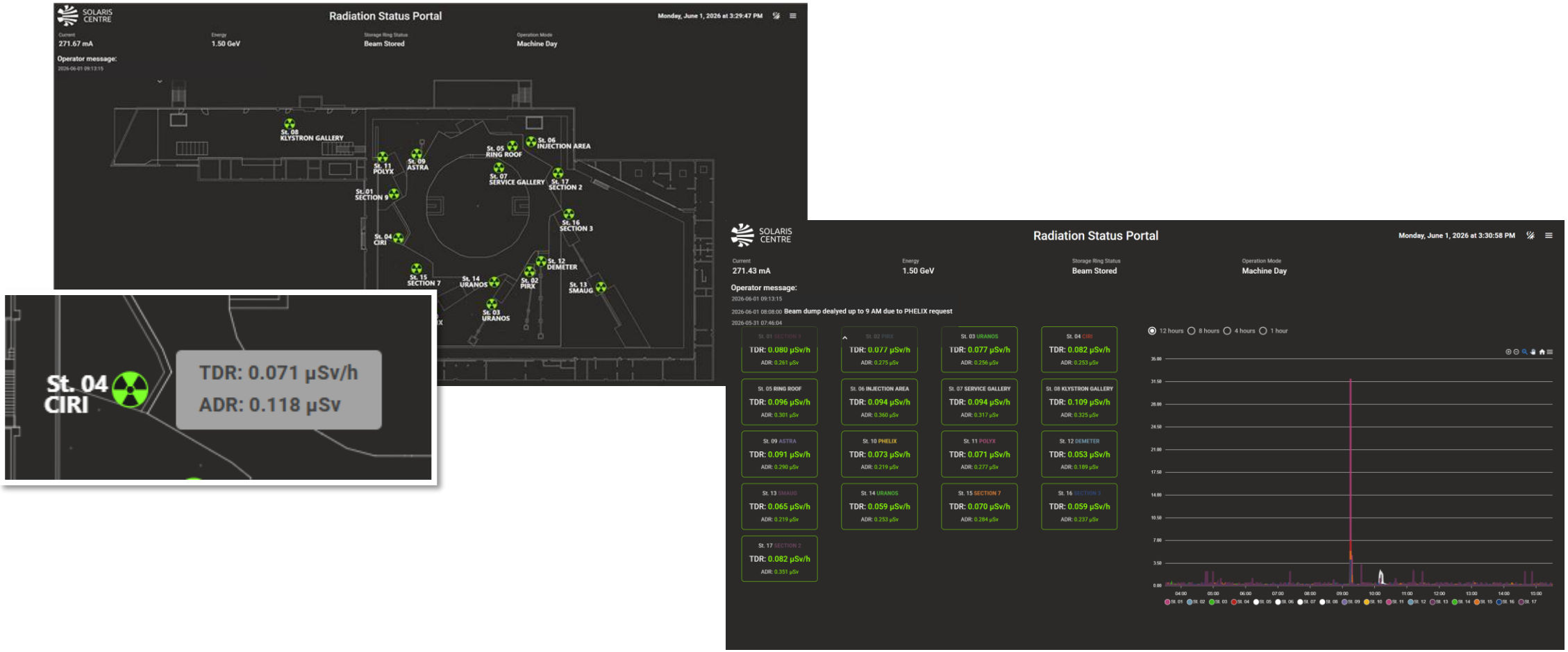
Radiation Status Portal

- Upgrade to Vue 3
- Introduce new Taranta Adriane with socket/query communication and Tango archiving for chart data



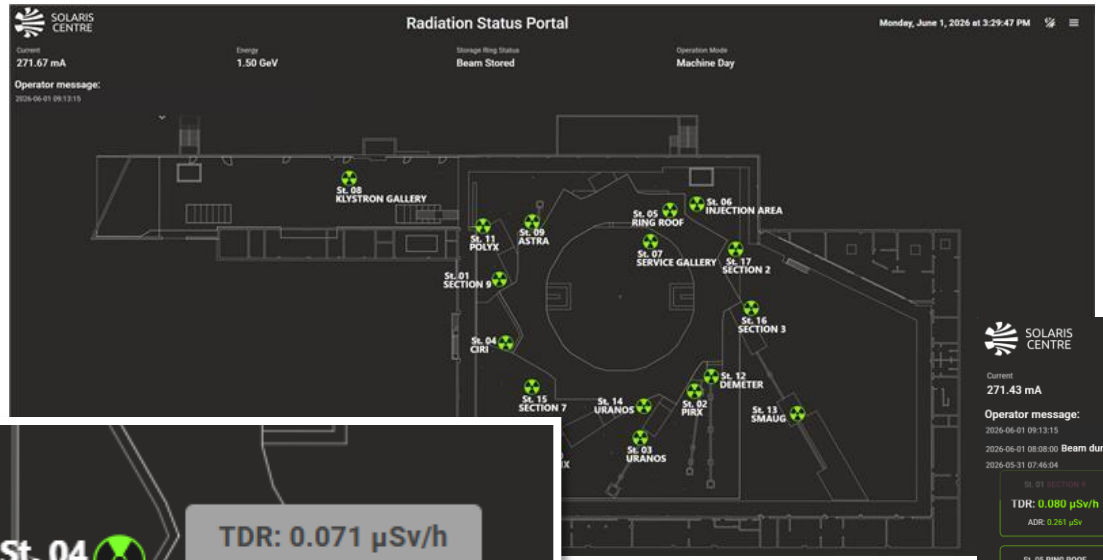
Radiation Status Portal

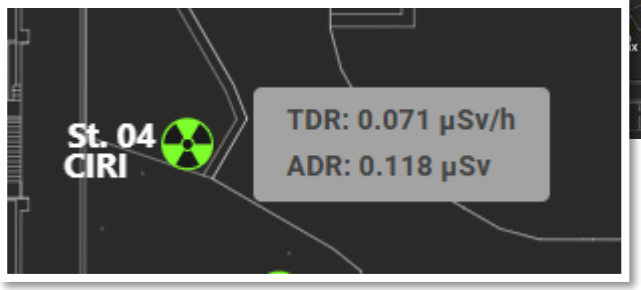
- Upgrade to Vue 3
- Introduce new Taranta Adriane with socket/query communication and Tango archiving for chart data
- Add a synoptic view (map-based interface) with station hover for quick value lookup

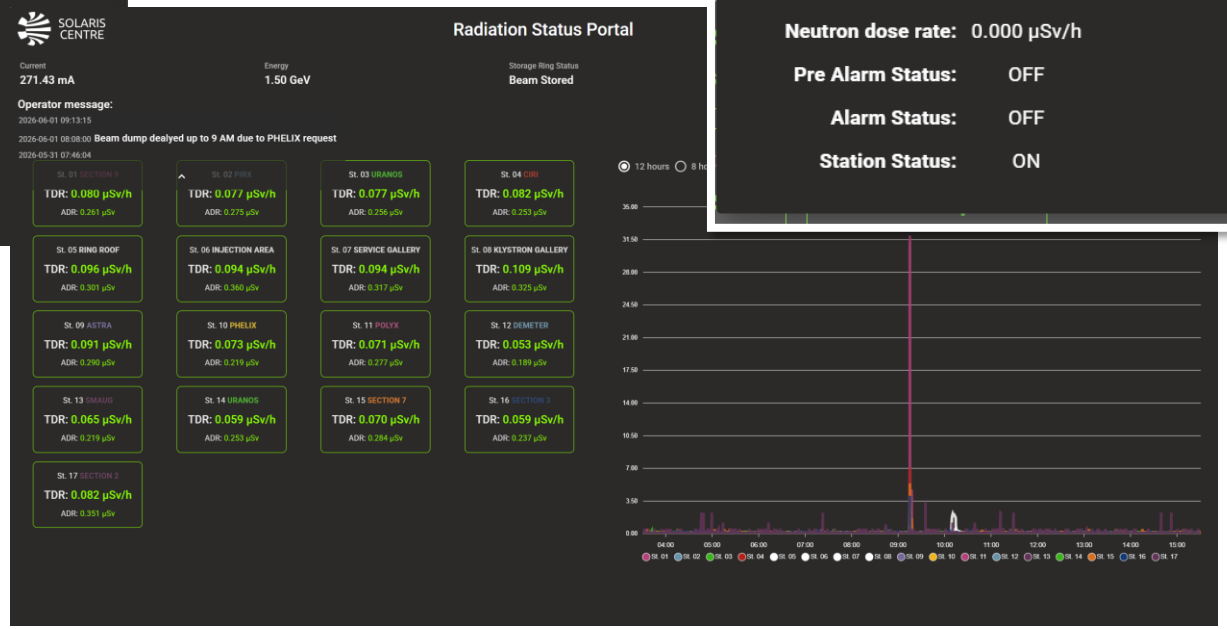


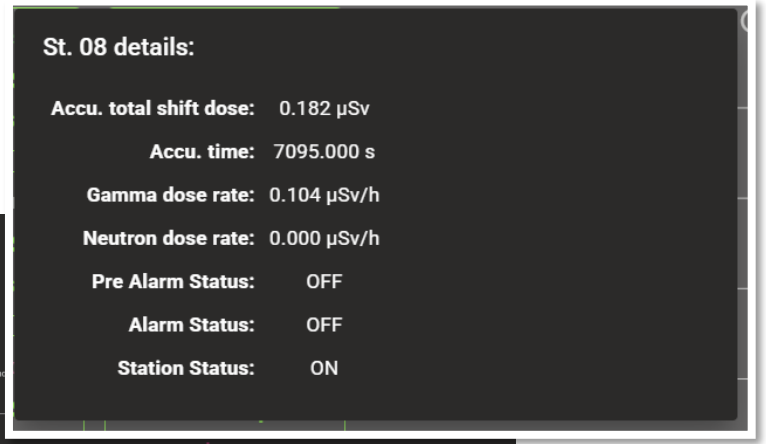
Radiation Status Portal

- Upgrade to Vue 3
- Introduce new Taranta Adriane with socket/query communication and Tango archiving for chart data
- Add a synoptic view (map-based interface) with station hover for quick value lookup
- Enable dialog display with more station's attributes (by clicking on stations or on a station card in the status view)









Thank you!

Questions?